

Noita Enola

Guía para aprender a
ESCANEAR

v1.0

No pre(a)ss(ure)

Índice

Índice	III
Prólogo	VII
I Preliminares	1
1 Introducción	3
1.1. Organización	3
1.2. Alcance de la guía	4
1.3. Sobre los formatos	5
2 Requisitos	7
2.1. Requisitos de software	7
2.2. Requisitos de hardware	13
3 Estructura del proyecto	19
II Procedimiento	21
4 Introducción	23

5	Escaneo	25
5.1.	Estado del aparato	25
5.2.	Libro	26
5.3.	Software	27
5.4.	A escanear	31
5.5.	Respaldo de los originales	35
5.6.	Conclusiones	36
6	Limpieza y ajuste	39
6.1.	Orientación de páginas	40
6.2.	Procesado con SCAN TAILOR	41
6.3.	Tapas	62
7	OCR con FINEREADER	65
7.1.	Configuración	66
7.2.	Creación del proyecto	70
7.3.	Reconocimiento	70
7.4.	Conclusiones	85
8	Creación del PDF	87
8.1.	Insertar imágenes	87
8.2.	Insertar marcadores	93
8.3.	Modificar etiquetas de páginas	95
8.4.	Metadatos y vista inicial	96
8.5.	Optimización	98
8.6.	Guardar en PDF/A	99
8.7.	Verificación	99
9	Creación del DJVU	101
9.1.	Configuración de perfiles	101
9.2.	Codificación con DJVU SMALL	103
9.3.	Codificación con DJVU SMALL MOD	110
9.4.	Insertar la capa OCR	114
9.5.	Insertar marcadores	117
9.6.	Modificar etiquetas de páginas	119
9.7.	Insertar metadatos	121
9.8.	Creación de hipervínculos	123

III Apéndices	125
A Otras herramientas útiles	127
B Conversión pix-pulgadas	129
C Sobre el material de ejemplo	131
D Alternativas para procesamiento de imágenes	133
D.1. Procesamiento posterior	134
E Materiales complejos	139
F OCR simplificado	141
F.1. Con ACROBAT DC	141
F.2. Con DJVU EDITOR	146
G Estructura del proyecto de FINEREADER 14	149
Referencias	153
Lista de figuras	155
Lista de cuadros	159
Lista de scripts	161
ToDo	163

Prólogo

El objetivo de esta guía es enseñar a digitalizar documentos impresos, especialmente libros.

No pretende ser la referencia definitiva, sino solamente mostrar un procedimiento válido y que en mi opinión alcanza buenos resultados.

A lo largo de diez años de escaneo aficionado fui aprendiendo, perfeccionando el proceso y tomando algunas notas. Últimamente he estado pensando que los conocimientos que he adquirido podrían llegar a ser de utilidad para otras personas. De mi propia experiencia y de esas notas es que surge esta guía, y de esas cavilaciones mi voluntad de compartirla. Espero que le sea de utilidad.

Noita Enola



CC BY 4.0

Parte I

Preliminares

Capítulo 1

Introducción

Bienvenido. En esta introducción comienzo con algunos comentarios generales, luego explico como está organizada la guía, que alcance tiene y hago un breve esbozo sobre los formatos.

Mucha de la información que aquí recopiló proviene de fuentes en ruso.¹ Desconozco las razones, pero tengo la impresión de que allí hay muchas más personas interesadas en el arte del escaneo de libros que en los países de habla hispana o inglesa. Una de ellas es monday2000² (monday2000@yandex.ru), que mantiene un importante sitio de referencia [1], con cantidad de información y software relacionado. Recomendando visitarlo.

Por supuesto existen otros sitios y personas que iré mencionando a medida que avancemos.

Otros materiales de referencia similares a este son [2, 3, 4, 5].

1.1. Organización

He decidido dividir la guía en tres partes:

¹ Para entenderlas he hecho amistad con el traductor de Google.

² Según dice al pie de su web, esta persona se llama en realidad Ivanov Victor. Desaparece de la escena alrededor del 2014.

Parte I La primera cubre los requisitos de software y hardware para comenzar a escanear. Hay una larga lista de programas con sus descripciones y una discusión sobre los distintos tipos de escáneres.

Parte II La segunda es la más extensa y contiene propiamente el procedimiento de digitalización, desde el escaneo hasta la producción de los documentos en distintos formatos. Expongo con cierto detalle cada paso y explico algunas alternativas de trabajo, tratando de resaltar las diferencias.

Parte III Para la última parte dejé los apéndices. Hacia aquí trasladé algunas discusiones que considero útiles pero que pienso hubieran entorpecido el flujo del proceso. También coloqué aquí las referencias.

1.2. Alcance de la guía

El método que presento funciona bien con impresos de texto y diagramas en negro sobre papel claro, con fotografías ya sea a color o escala de grises.

Algunas características que tal vez no sea posible reproducir de forma balanceada simplemente siguiendo esta guía son:

- texto en múltiples colores (todo se pasará a negro);
- páginas con fondo de color (si es suficientemente claro se convertirá en blanco, de lo contrario dará problemas); y
- páginas con imágenes de fondo (parecido al caso anterior, pero peor).

Estas limitaciones son muy reales, especialmente en la digitalización de folletos, revistas, cómics o libros a todo color. Para aquellos casos, espero que esta guía pueda ser un buen punto de partida desde el que cada uno pueda desarrollar sus propias técnicas y decidir si un procedimiento es más satisfactorio que otro en función de la calidad, el tamaño final del archivo, el esfuerzo requerido, etc. En el apéndice E esbozo un procedimiento simple para tratar este tipo de materiales.

En el caso de los libros, que es el que más me interesa, existen multitud de publicaciones que aún no han sido digitalizadas y que caen

dentro del alcance de esta guía, y desde allí pienso que esta puede ser un aporte valioso. Para muchos otros existen versiones digitales nativas (PDF vectorial), por lo que no será necesario escanear nada.

1.3. Sobre los formatos

Si todo sale bien, al final del proceso de digitalización deberíamos tener una copia digital del material físico pero, ¿en qué formato?

pdf La realidad es que desde hace ya muchos años el formato dominante ha sido el PDF. No me detendré a analizar las razones de esto, simplemente diré que, a pesar de que funciona bien, *no* es el formato más adecuado para esta función. Principalmente porque no fue diseñado para esto. Su principal ventaja es su soporte prácticamente universal. Sus desventajas son que para una calidad similar los archivos son mucho más grandes y la visualización en pantalla es sensiblemente más lenta.

djvu El formato DJVU fue diseñado específicamente para el almacenamiento y distribución de documentos escaneados e imágenes de alta resolución [6]. Utiliza el modelo MRC (*mixed raster content* o contenido rasterizado mixto), definido en la norma ISO/IEC 16485 [7]. Simplificando un poco,³ en este modelo la imagen se descompone en capas de primer plano y fondo, y se aplican distintas técnicas de compresión optimizadas para cada una. Sus ventajas incluyen la sencillez del formato,⁴ el reducido tamaño de los archivos resultantes (para una misma calidad) y la velocidad de visualización. Las desventajas principales son la poca aceptación que ha tenido [10] y la escasez de herramientas para la creación y edición de archivos.

Por supuesto estas ventajas y desventajas dependen de muchos otros factores, algunos de ellos subjetivos, por lo que no pretendo zanjar aquí la cuestión. Personalmente prefiero DJVU para documentos escaneados. Por más información se pueden consultar [11, 12, 13, 14, 15, 16, 17] y cualquier buscador en línea.

No es necesario seguir todos los pasos de la guía para obtener los documentos en ambos formatos. Podría producirse solo el DJVU y

³ Para más información pueden consultarse las secciones 3.1 y ss. de [8].

⁴ 70 páginas en la referencia de DJVU [8] contra 750 páginas en la de PDF [9].

luego convertirlo a PDF con alguna herramienta, para luego hacer algún retoque, por ejemplo. El camino inverso también es posible pero estimo que el resultado no será tan bueno.

Requisitos

Los requisitos de software y hardware varían según el resultado que se esté buscando. A continuación se listan más o menos exhaustivamente, aunque un ítem u otro podrían ser opcionales de acuerdo a las necesidades.

Además, hay un cierto requisito humano: es necesario saber desenvolverse con cierta soltura frente a un PC. Esta guía trata de ser completa, pero siempre surgirán problemas, casos no previstos, incompatibilidades, etc. Cuando esto suceda relea la guía, lea la documentación relacionada con lo que no está saliendo bien, pida ayuda en internet. ¡No se desanime, estimado lector!.

2.1. Requisitos de software

Primero que nada, soy un usuario de Windows de toda la vida, por lo que todo el trabajo lo hago sobre Windows 7. Muchos de los programas que siguen tienen versiones para Linux, aunque no todos. Lamentablemente, no existen alternativas libres para cada uno, o al menos que produzcan un resultado de igual calidad. Todo esto podría cambiar en el futuro.

La lista de programas que sigue es bastante extensa por la razón de que he querido mencionar las distintas alternativas disponibles. Ciertamente no será necesario instalarlos todos.

2.1.1. Variable PATH

Recomiendo instalar todos los programas en una carpeta de fácil acceso, por ejemplo  C:\SOFT. Además, como invocaremos varios programas usando scripts, será importante que la carpeta donde se instaló cada uno esté incluida en la variable `PATH` del sistema [18, 19].

2.1.2. Generales

Estos son programas de uso general. Algunos asociados con el correcto funcionamiento de diferentes aplicaciones y otros con el manejo de archivos de todo tipo.

VirtualBox Software de virtualización. Principales funciones bajo licencia GPLv2. Se trata de un programa completamente opcional, pero no es mala idea tener todo el flujo de trabajo encapsulado en una máquina virtual. En caso de usarse, todo el software siguiente se instalaría en la VM creada.

Drivers Drivers del escáner correctamente instalados y funcionando.

.NET Framework Runtime No sabría definirlo. Licencia MIT. Es necesario para correr algunos programas, en particular el DOCUMENT EXPRESS. Podría ser opcional si no se instala este último.

Java Ídem anterior. Freeware. Necesario si se va a usar VERAPDF.

Python 3 Lenguaje de programación sencillo (para lo que lo vamos a usar). Licencia PSFL (estilo BSD). Necesario para ejecutar los scripts que tengo preparados.

7-Zip Archivador de ficheros. GPLv2+BSD. Buena alternativa a WIN-ZIP y WINRAR. Desde que lo empecé a usar no he mirado atrás. Opcional una vez que esté todo funcionando.

Advanced Renamer Renombrador de archivos masivo. Freeware. Podría no llegar a necesitarse. Hay mil alternativas.

Hasher Generador de hashes. Freeware. Hay miles de alternativas. Totalmente opcional.

HEXelon MAX Calculadora, conversor de unidades y monedas, etc. Freeware. Alternativa a la calculadora de Windows. Hay otras. Podría sustituirse por una calculadora de escritorio.

2.1.3. Visualizadores de documentos

Programas para visualizar archivos de texto, imágenes y documentos. También nos servirán para escanear nuestro material.

Notepad++ Editor de texto y código fuente. Licencia GPLv2. Adiós Notepad de Windows.

IrfanView Visor de imágenes freeware. Será utilizado para la adquisición de imágenes desde el escáner.

XnView Visor de imágenes freeware. Es con el que me siento cómodo para visualizar imágenes en general. Existen muchas alternativas disponibles (por ejemplo, IRFANVIEW).

SumatraPDF El mejor visualizador de documentos multiformato. Punto. Licencia GPLv3. Para ver los PDF y DJVU que se vamos a generar.

WinDjView Visualizador de archivos DJVU. GPLv2. El SUMATRAPDF lo suple en general bien, pero este muestra alguna información adicional específica que puede sernos útil.

2.1.4. Procesadores de imágenes

Se trata de software para procesar las imágenes escaneadas, realizar algún retoque o edición por lotes.

Scan Tailor Advanced Herramienta interactiva para el postprocesamiento de las páginas escaneadas. Licencia GPLv3. Núcleo de todo el proceso. Con este software se enderezarán y binarizarán las páginas, ajustarán los márgenes, separarán las imágenes, etc. Es un fork del original **SCAN TAILOR**, prácticamente abandonado desde el 2012. Existe otro fork similar y aún mantenido: **SCAN TAILOR UNIVERSAL**. No tengo experiencia con él.

ScanKromsator Alternativa a **SCAN TAILOR**. Aun más completo. Por lo que entiendo existe desde bastante antes y es muy usado en Rusia. Lamentablemente posee escasa documentación y ésta es poco actualizada. Puede ser difícil conseguir información o soporte en otro idioma que no sea ruso.

gimp Programa de edición de imágenes. Licencia GPLv3. Es una alternativa libre a Photoshop. Se usará para retocar detalles de las páginas escaneadas.

ImageMagick Conjunto de utilidades para manipular y convertir imágenes. Licencia Apache 2.0. Lo usaremos para procesar imágenes por lotes.

2.1.5. Generadores de ocr

Este será el software que utilizaremos para reconocer el texto en nuestros documentos escaneados.

ABBY FineReader Aplicación de software de OCR (reconocimiento óptico de caracteres). Software propietario y de pago. Lo usaremos para insertar una capa de texto seleccionable a nuestros documentos finales.

tesseract Motor de OCR libre para varias plataformas. Licencia Apache 2.0. Hay varias interfaces gráficas disponibles. No tengo experiencia con él. Para usarlo habría que hacer adaptaciones al flujo de trabajo que viene más adelante.

Nuance Omnipage Alternativa propietaria y paga. Más bien orientada al reconocimiento en serie de grandes volúmenes de documentos. Tampoco la he probado. No estoy seguro de que maneje archivos DJVU.

2.1.6. Productores de documentos

Este software es el que nos permitirá generar y editar nuestros documentos a partir de las imágenes procesadas.

Adobe Acrobat DC Permite visualizar, crear y modificar archivos PDF. También hace OCR. Software propietario y de pago. Lo usaremos para personalizar los marcadores, los números de página, insertar las imágenes y generar documentos PDF/A, para el guardado a largo plazo de documentos electrónicos. Las versiones anteriores también servirán, salvo quizás para el paso de insertar imágenes (se requiere la ejecución de javascripts de Adobe, que no he probado con estas). Dependiendo del resultado que busquemos, podría ser opcional.

DjVuLibre Implementación libre del formato DJVU. Incluye visor, decoder, algunos encoders simples y otras herramientas varias. Licencia GPLv2. Si bien podría emplearse para generar nuestros documentos DJVU, de este paquete usaremos casi exclusivamente las herramientas de manipulación ( `djvused.exe`,  `djvm.exe`, etc.).

LizardTech DjVu Document Express Enterprise Aplicación comercial para la generación de archivos DJVU. Software propietario y de pago. Discontinuado en 2006.¹ La alternativa libre es DJVULIBRE. Sin embargo, la calidad de los encoders de LizardTech es superior salvo, tal vez, para la codificación de fotografías.² Si bien hay

¹ AT&T es dueña de varias patentes relativas al formato. (Para más información, ver [20, 21].) Desde hace algunos años los derechos de uso comercial de estas se han ido transfiriendo a diferentes compañías: LizardTech (ahora Extensis), Celartem (que adquirió estas dos) y Cuminas. Esta última ofrece un producto con el mismo nombre pero completamente distinto.

² De la web de DJVULIBRE: «The compressors provided here are slower, produce larger files (sometimes with lower image quality) than the commercial compressors, but they do the job».

versiones más recientes, la v5.1.0 (b946 o b973) funciona bien. No lo usaremos directamente, por lo que no hay necesidad de conseguirlo.

LizardTech DjVu Editor Creador y editor de ficheros DJVU. Software propietario y de pago. Tiene una interfaz amigable que permite insertar páginas en un documento, así como crear hipervínculos. Incluye OCR. Casi opcional.

DjVu Small Interfaz de usuario para las aplicaciones de línea de comandos `Ⓜdjvudecode.exe` y `Ⓜdocumenttodjvu.exe` de DOCUMENT EXPRESS (v5.0.0.973). La licencia de la GUI es GPLv2, pero las aplicaciones que controla son comerciales. Su autor es monday2000. Sigue muy vigente a pesar de que se dejó de desarrollar en 2006.³

DjVu Small Mod Fork de DJVU SMALL desarrollado por textsharik (textsharik@ya.ru) a partir del 2011. Es un proyecto bastante más ambicioso, que ya incluye la funcionalidad de DJVU IMAGER, la separación automática o semiautomática de capas,⁴ agrega varias modificaciones más a `Ⓜdocumenttodjvu.exe` (por ejemplo, procesamiento multihilos) e incluye ejecutables de versiones más nuevas de DOCUMENT EXPRESS (para convertir de DJVU a PDF). Licencia GPLv2. Incluye completa documentación en ruso. Pasándola por el traductor de Google se entiende bastante bien.

2.1.7. Herramientas varias

Sobre todo para el formato DJVU, existen varias herramientas que vamos a necesitar en los distintos pasos.

qpdf Aplicación de línea de comandos que hace transformaciones estructurales y que preservan el contenido de archivos PDF. Licencia Apache 2.0. Lo usaremos para insertar las imágenes en nuestro documento PDF.

³ El autor ofrece por un lado los fuentes, que son libres, y por otro los binarios, que ya incluyen las aplicaciones comerciales. En particular, `Ⓜdocumenttodjvu.exe` fue modificada para funcionar sin licencia y redirigir ciertos mensajes internos hacia la consola, lo que permite visualizar el proceso de codificación en tiempo real.

⁴ No las cubro en esta guía.

DjVu Imager Herramienta para pegar imágenes en archivos DJVU. Licencia GPLv2. La necesitaremos si vamos a utilizar el método de la sección 6.2.6.

PdgCntEditor Editor de marcadores para PDF y DJVU. Freeware para uso no comercial. Brinda una cantidad de herramientas para la generación y edición de marcadores en estos dos formatos.

bookm Herramienta desarrollada por monday2000. A partir de un archivo de marcadores convencional genera otro con la estructura necesaria para usar con  `djvused.exe`.

2.1.8. Verificadores

En el caso del formato PDF, existe un software que permite verificar que nuestro documento cumple con el estándar PDF/A de almacenamiento de largo plazo.

veraPDF Herramienta de validación de documentos PDF/A. Licencia GPLv3 o MPLv2. La usaremos para verificar el PDF/A que hayamos creado.

2.2. Requisitos de hardware

Los requisitos de hardware son bastante más escasos, pero hay una discusión interesante sobre el escáner.

2.2.1. Material para digitalizar

Esta guía está enfocada en la digitalización de libros, pero su lectura podría resultar útil para la digitalización de otro tipo de documentos.

2.2.2. Regla de longitud suficiente

Precisión a gusto del usuario. Es un elemento opcional que usaremos para hacer coincidir las medidas reales del impreso físico con las del documento final. Debe poder medir al menos la dimensión más larga

del material. Puede omitirse en caso de que esta coincidencia no sea de interés.

2.2.3. Escáner

Respecto al escáner, existen multitud aspectos a tener en cuenta. El correspondiente artículo de la Wikipedia [22] es un buen punto de partida. A continuación se esbozan algunas características de las principales tecnologías. Por supuesto, existen otras.

Escáner de libros

Para la digitalización «profesional» de documentos existen multitud de opciones, sobre todo orientadas a bibliotecarios y archivistas. Son los así llamados *book scanners*, *large format scanners*, *document cameras*, etc. Consisten generalmente en una superficie sobre la que se coloca la obra a digitalizar, iluminación adecuada y una o más cámaras en la parte superior, que van tomando fotografías de alta resolución. Luego, mediante software incluido realizan correcciones a las imágenes, reconocen el texto, generan documentos en varios formatos, etc. Todo esto de manera rápida, automática o semiautomática, pues están pensados para el procesamiento de grandes volúmenes de ejemplares.

Son aparatos generalmente costosos y voluminosos. En la figura 2.1 se ilustran algunos ejemplos.

Es de destacar la labor que desde hace años viene realizando la comunidad DIY Book Scanner [23], fundada por Daniel Reetz, para desarrollar soluciones de escaneo accesibles para cualquiera. El sitio y sus foros contienen cantidad de información útil sobre cada etapa tanto de la construcción de un escáner (fig. 2.2) como del posprocesamiento de las imágenes, entre muchos otros temas relacionados.

Otros proyectos similares y aún en desarrollo son Libreflip [24], de Ijon Berlin, y Linear Book Scanner [25], de Dany Qumsiyeh.

Escáner de cama plana

Estos son los más comunes y económicos. Existen básicamente dos tecnologías: CIS y CCD. Hay una cantidad interesante de detalles técnicos, pero básicamente la tecnología CCD es un poco mejor aunque algo



(a) ScanSnap SV600, de Fujitsu.



(b) ET16, de CZUR.

FIGURA 2.1. Dos ejemplos de *book scanners*.



FIGURA 2.2. Archivist. Desarrollado por Daniel Reetz con ayuda de la comunidad *DIY Book Scanner*.



(a) Vista general.

(b) Detalle. Nótese que el vidrio alcanza casi el borde.

FIGURA 2.3. **OpticBook 3800**, de **Plustek**.

más cara. En mi experiencia, cualquiera de los dos funciona suficientemente bien con una gama amplia de materiales mientras se acerque lo suficiente la página al vidrio del escáner [26].

Este tipo de escáneres son los que ofrecen las impresoras multifunción, aunque también se venden por separado.

Existe una subvariedad en esta categoría, diseñada específicamente para escanear libros. La característica saliente es que el vidrio del escáner llega casi al borde del aparato, lo que permite alcanzar fácilmente el espacio en donde se unen las páginas. Ver ejemplo en la figura 2.3.

Escáner con adf

ADF significa alimentador automático de documentos (*automatic document feeder*, en inglés). Estos escáneres son muy prácticos, pues toman cada hoja de una pila y la escanean sin necesidad de intervención humana, algunos inclusive por ambas caras. La desventaja es que las hojas a escanear deben estar sueltas. Son muy comunes en las oficinas. Algunos ejemplos en la figura 2.4.



(a) Workforce DS-410, de Epson.



(b) ScanJet Pro 2500, de HP.

FIGURA 2.4. Escáneres con ADF.

Estructura del proyecto

Los archivos de nuestro proyecto estarán estructurados en un árbol de carpetas como el de la figura 3.1. Habrá una carpeta base llamada **Project**, dentro de la que iremos creando las siguientes. Esta carpeta base puede estar en cualquier lugar de nuestro PC. Normalmente la coloco en la raíz del disco **C:**, pero los scripts que utilizaremos deberían funcionar sin importar donde esté ubicada, mientras se respete la estructura del árbol.¹

En **SCAN** irán nuestras exploraciones hechas con el escáner. Dentro de esa carpeta tendremos **out**, donde **SCAN TAILOR** guardará

¹ Hay excepciones que se mencionan específicamente.

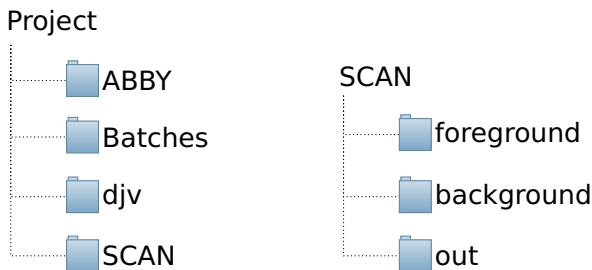


FIGURA 3.1. Árbol de directorios del proyecto.

las imágenes procesadas. A su vez, podrían estar  foreground y  background, donde se guardarán independientemente el texto y las imágenes.

En  Batches guardaremos los scripts y algunos archivos asociados a ellos.

En la carpeta  ABBY irá nuestro proyecto de FINEREADER. Dentro de esa carpeta el programa creará muchas otras, con la estructura que se describe en el apéndice G.

En  djv guardaremos varios documentos PDF y DJVU, que corresponderán a las distintas etapas de procesamiento.

Parte II

Procedimiento

Introducción

En esta parte se explica cada paso del proceso hasta la producción de los archivos digitales.

Como ejemplo vamos a utilizar un libro de instrumentación y medidas de B. A. Gregory [27].¹

Comenzaremos explicando los trabajos de escaneo del material original: preparación del escáner, preparación del material, configuración del software de escaneo, proceso de escaneo y posibles complicaciones.

Luego pasaremos al procesamiento de las imágenes adquiridas, para el que utilizaremos fuertemente el software SCAN TAILOR.

El siguiente paso será reconocer el texto con FINEREADER. Para ello deberemos configurarlo adecuadamente y habrá una cantidad de obstáculos que tendremos que evitar.

Por último crearemos los documentos finales en formato PDF y/o DJVU. Deberemos agrupar las imágenes en estos archivos y agregarles el texto reconocido, las imágenes, los marcadores, etiquetas de páginas y metadatos.

Sin más dilaciones: ¡comencemos!.

¹ Para más información, ver el apéndice C.

Capítulo 5

Escaneo

La regla número uno para obtener un buen resultado es comenzar con buen material de base. En el caso que nos ocupa necesitaremos imágenes de buena calidad, es decir: de adecuada resolución, almacenadas en el formato adecuado, usando el algoritmo adecuado, sin artefactos, etc. En esta sección veremos algunas recomendaciones para conseguir las. Recomendando antes visitar [28].

5.1. Estado del aparato

Lo primero que debemos hacer es revisar el escáner que vayamos a usar. Para ello puede ser útil tener a mano una linterna. El vidrio debe estar limpio, libre de grasa, vellos, polvo y cualquier otro tipo de suciedad. Lo mismo la parte interna de la cubierta del escáner, pues aunque muchas veces no se use podría dejar caer suciedad sobre el vidrio. Si no lo están, deben primero limpiarse según las especificaciones del fabricante. En caso de que no existan, un paño de microfibra o papel absorbente apenas humedecido con agua (o alcohol isopropílico) podría funcionar. Si el vidrio se encuentra rayado conviene hacer alguna exploración de prueba para ver si la afecta apreciablemente. De ser así deberemos evitar esa zona del escáner y, en caso de no poder hacerlo, procurar otro aparato.

Además, deben haberse instalado correctamente los drivers y el

software de digitalización. En caso de tener problemas con esto, lo mejor será consultar en la web del fabricante o algún foro de soporte en internet.

5.2. Libro

¿Hay que preparar el libro?! Bueno, eso dependerá de varias cosas.

Lo primero que debemos preguntarnos es que valor (sentimental o material) tiene lo que vamos a escanear, y cuanto de ese valor estamos dispuestos a sacrificar o a poner en riesgo.

Si estamos hablando de un libro barato o al que no nos sentimos apegados, la opción más eficiente es cortarle el lomo en una guillotina¹ y colocarlo en un escáner ADF. Si no tenemos ese tipo de escáner, de todas formas es más fácil colocar las hojas sueltas en uno de cama plana y conseguir digitalizaciones de muy buena calidad. Luego de esto se pueden tirar las hojas o, si vale la pena, enviarlas a reencuadernar.

Si el material no puede ser destruido, debemos preguntarnos si aún así no puede ser abusado. Es decir, ¿podemos permitirnos abrirlo 180° y apretarlo con fuerza contra el vidrio del escáner? He escaneado muchos libros de esta manera y salvo algunas excepciones nunca han perdido páginas ni quedado con daños permanentes, pero esa evaluación la debe hacer cada persona. Aún si estamos dispuestos, podría ocurrir que no logremos buenos resultados en la parte donde se unen las hojas (ver figura 5.2). Para estar seguros podemos hacer algunas previsualizaciones de prueba.

Si seguimos sin encontrar la solución, debemos considerar adquirir otro tipo de escáner. O tal vez otro ejemplar de la obra que tenga menos valor y permita estos abusos.

Los productos que fabrica para esto Plustek funcionan bastante mejor, pero igual se debe hacer algo de presión sobre el lomo para que toda la imagen sea clara, sobre todo en volúmenes muy gruesos.

Luego están los *book scanners* comerciales, pero por estar pensados para materiales que no pueden ser abusados, dependen mucho del software que traen para corregir las aberraciones ópticas que tiene la imagen, producto de que la página nunca se encuentra plana respecto a las cámaras.

¹ O cortar sus páginas de una, para los que tengan paciencia.

Un punto intermedio es el Archivist o similares, que abren los libros solo unos 135° y suavemente presionan una cuña de vidrio para aplanar las páginas.

No se desanime si su caso no está cubierto en estas líneas. Piense, experimente, pregunte. Esta guía no lo cubre todo y la solución que necesita para su proyecto puede estar a la vuelta de la esquina.

5.3. Software

El paso siguiente es configurar el software de escaneo. Generalmente, este viene pensado para que se pueda usar sin pensar demasiado y que aún así puedan obtenerse buenos resultados. Por ello ya se incluyen cantidad de herramientas de posprocesado (mejora de la nitidez, desinclinación, balance de blancos, separación de varias imágenes, entre muchas otras). Debemos desactivarlas todas. Tengamos en cuenta que de ser necesarias algunas de estas correcciones podremos realizarlas más adelante y con mucho más control utilizando programas adicionales.

Si abrimos el IRFANVIEW y elegimos **File >> Acquire/Batch Scanning...** obtenemos el cuadro de diálogo de la figura 5.1(a). Debemos seleccionar **☒ Multiple images (Batch mode)** y elegir la opción **Save as >> TIF (Tagged Image File Format)** del menú **Save as** y luego presionar **Options**. Allí aparece el cuadro de abajo, donde se debe seleccionar el tipo de compresión. Sin entrar en más detalles, elegimos **☒ LZW**, pues no tiene pérdidas (*lossless*) y ofrece un grado razonable de compresión. Otra alternativa es **☐ ZIP**, pero alguna vez me dio problemas con el SCAN TAILOR². Por ninguna razón elegir **☐ JPEG**, pues introduce pérdidas (*lossy*) o **☐ None** (sin compresión) porque genera archivos enormes.³ Las demás opciones pueden elegirse como en la figura o cambiarse a gusto.

Luego de elegir las opciones presionamos **OK** y si el escáner está bien configurado debería iniciarse el software correspondiente. Como ejemplo usaremos el programa EPSON Scan (v5.7.9.3). En la figura 5.2

² Esa realidad pudo haber cambiado. No he probado **Packbits**.

³ Nuestro proceso contendrá muchos archivos temporales de gran tamaño que al final terminarán en un documento de tamaño razonable. Estos archivos temporales podrán ser borrados al terminar, pero si de todas formas pudiéramos ahorrar algo de espacio durante el proceso lo haremos.

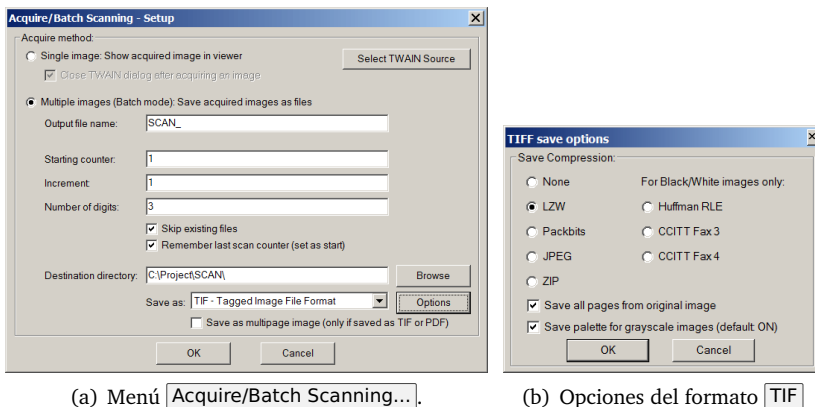


FIGURA 5.1. Ajustes de escaneo del IRFANVIEW.

se muestra la ventana principal. Podemos ver que se ofrecen varios **Modo**: **Automático**, **Fácil**, **Oficina** y **Profesional**. Por supuesto elegimos **Profesional**.

Comenzamos por colocar una página representativa del material a escanear y luego presionamos **Previsualización**, lo que nos dará un resultado como el de la fig. 5.2. En caso de que la región a escanear no ocupe toda la superficie del escáner, conviene seleccionarla dibujando un rectángulo en la pestaña de **Previsualización**. De esta forma se agilizan un poco los tiempos de escaneo.

En nuestro caso, las opciones **Tipo Documento(Y)** y **Origen Documento** no se pueden modificar. **Tipo Auto Exposición** se explica más abajo.

En **Tipo de imagen** elegimos **Color 24-bit** o **Grisés 8-bit**, dependiendo de si la imagen es a color o no. Se ofrecen también **Suavizado de Color** y **Ninguno**. Según la documentación de Epson la primera es para «gráficos a color, como un diagrama o un gráfico». La segunda no tiene un nombre muy explícito pero significa bitonal. Es decir, blanco o negro. Esta opción puede ser tentadora porque generalmente produce el tamaño de imagen más pequeño. Sin embargo, se pierde muchísima información de la imagen (ver figura 5.3). Nuestro resultado final será en general bitonal, pero luego de todo el proceso tendremos imágenes

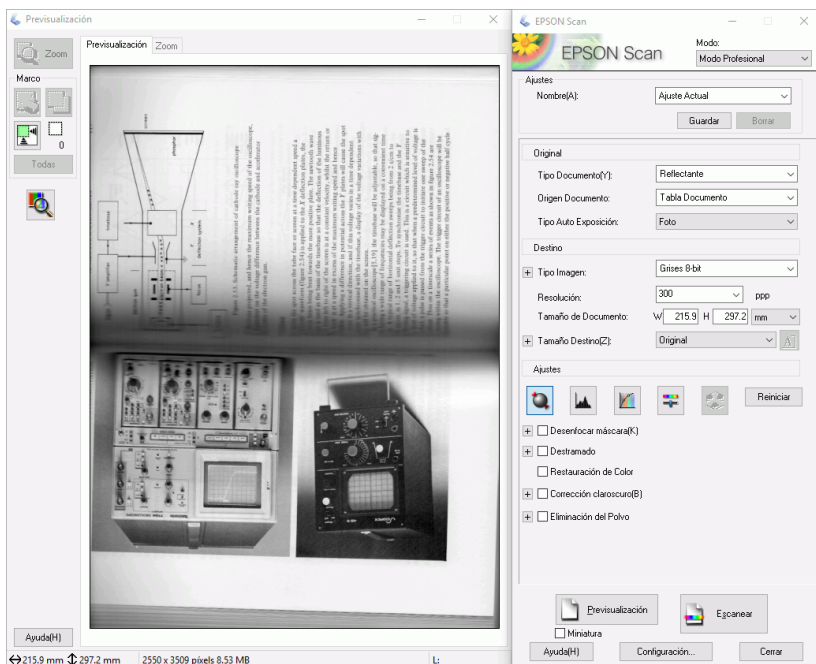


FIGURA 5.2. Ventana principal del EPSON Scan. Se previsualizó una página representativa.

de excelente calidad e ínfimo tamaño. Por todo esto, no debe usarse ninguna de estas dos opciones u otras similares que puedan ofrecer las aplicaciones de los distintos fabricantes.

La **Resolución** es uno de los parámetros más importantes. La óptima es de 300 dpi (ppi o ppp). Así se obtiene un buen resultado final y no se hace eterno el tiempo que se está presionando el libro contra el escáner. 600 dpi sería aún mejor, pero en los escáneres comunes cada exploración toma mucho más tiempo. Valdrá la pena solo para páginas con un nivel muy alto de detalles. Si bien la elección final dependerá de cada caso, escanear a menos de 300 dpi será siempre un desperdicio, por la cantidad de detalles que se perderán.

Tamaño de Documento se ajustará a la región que hayamos seleccionado (si lo hicimos) y **Tamaño de Destino(Z)** debe decir **Original**.

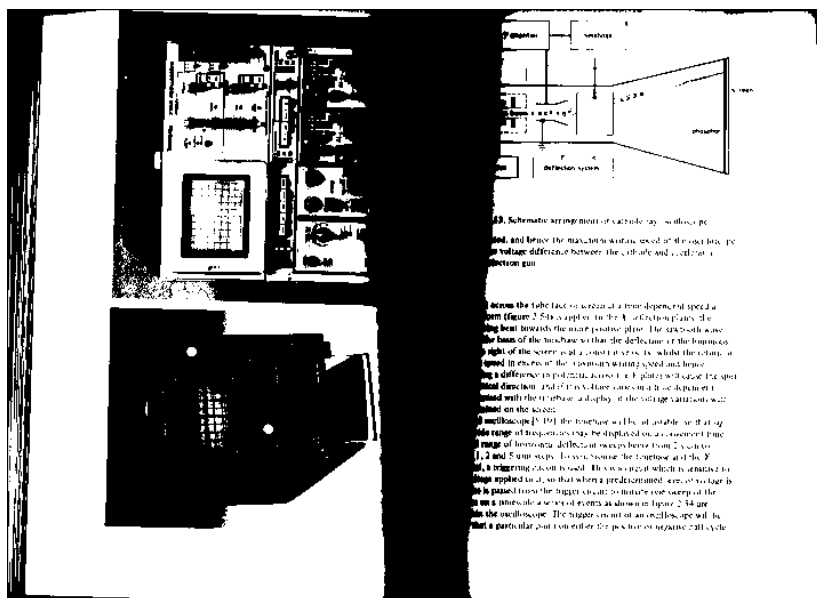


FIGURA 5.3. La misma imagen que en la fig. 5.2 pero habiendo seleccionado la opción **Ninguna**. Nótese que las fotografías están arruinadas y que el texto es ilegible.

Como se muestra, recomiendo la desactivación de todas las casillas:

☐ Desenfocar máscara(K), ☐ Destramado, ☐ Restauración de Color, ☐ Corrección claroscuro(B) y ☐ Eliminación del Polvo, por las razones aducidas al principio de esta sección.

En la parte de **Ajustes** hay una serie de íconos que representan distintos tipos de ajustes que se pueden aplicar. Al presionar **Previsualización** o cambiar el área seleccionada se activa automáticamente el primero, que es el de **Autoexposición**. Este ajusta a todos los demás en función de la imagen previsualizada (o el área seleccionada) y la opción **Tipo Auto Exposición** elegida.

Normalmente utilizo la opción de **Autoexposición** o presiono **Reiniciar** y dejo todos los valores por defecto. Alguien con conocimientos de ajuste de imágenes podría hacer un **Ajuste de Histograma**, **Corrección de Tono**, **Ajuste de Imagen** (brillo, contraste, saturación, balance de colores) o

modificación de la `Paleta de Colores` (solo para `Color 24-bit`) y es libre de intentarlo. Sin embargo, debe tenerse en cuenta que el ajuste que funciona muy bien para una página podría no funcionar para todas las demás, por lo que yo limitaría el ajuste fino a páginas específicas, y para las demás usaría los valores por defecto o el autoajuste con una página representativa (de manera de que funcione más o menos para todas las demás). En caso de dudas lo mejor es experimentar.

Para este proyecto utilicé una misma configuración para todas las páginas, tuvieran imágenes o no.

5.4. A escanear

Una vez preparado todo, podemos comenzar a escanear. Para ello presionamos `Escanear`. El equipo hará una exploración y el IRFANVIEW irá mostrando y guardando las imágenes de acuerdo a las opciones seleccionadas. Lo hará usando el nombre base (`Output file name`) y aumentando la numeración. Es importante que el orden creciente de las páginas del libro corresponda con el orden creciente de los números de los archivos, para que no se nos haga lío a la hora de compaginar todo. Además, algunos scripts y aplicaciones que usaremos luego dependerán de que esta numeración sea consistente a lo largo de todo el proyecto para funcionar correctamente.

Si el libro es suficientemente pequeño podremos ir de a dos páginas a la vez, de lo contrario será de a una. Luego de un rato de estar escaneando veremos que ya lo hacemos prácticamente sin pensar. Llegado ese punto yo aprovecho para mirar televisión, escuchar música o meditar sobre lo humano y lo divino.

Conviene que las páginas estén rectas respecto a los bordes de la superficie de exploración, pero sin necesidad de perfección. Durante el procesamiento podremos corregir algunas desviaciones.

Es buen momento para poner sobre la mesa de cuanto tiempo estamos hablando. En el caso de este libro usé un escáner OpticBook 3800 configurado en 300 dpi y grises 8-bit. Escanéé las primeras 200 páginas de un tirón (una página por exploración) y me llevó 40 minutos. Estamos hablando entonces de un promedio de 5 pág/min. El libro tiene unas 450 páginas por lo que en total debería llevar al menos 90 minutos.

Por supuesto que luego hay que revisar y reescanear las que no hayan quedado bien.

Al finalizar deberíamos tener aproximadamente tantos archivos como páginas tiene el libro (salvo que hayamos adquirido de a dos). Conviene hacer una inspección visual de cada imagen para ver si alguna no quedó del todo bien y necesita escanearse de nuevo. A continuación veremos algunos de los problemas que pueden surgir y como remediarlos.

5.4.1. ¿Qué partes escanear?

Antes de continuar hay un aspecto importante que no quisiera dejar de resaltar. En general, pero sobre todo si se trata de un libro publicado, escanéelo de cubierta delantera a cubierta trasera, sin dejar fuera absolutamente nada: ni páginas en blanco (llamadas de respeto o de cortesía), ni páginas de copyright, ni índices, ni apéndices, ni lomo... Todo tiene su razón para estar allí y hace a la calidad de la publicación (y, por ende, de la digitalización), por lo que no hay motivo para obviar nada.

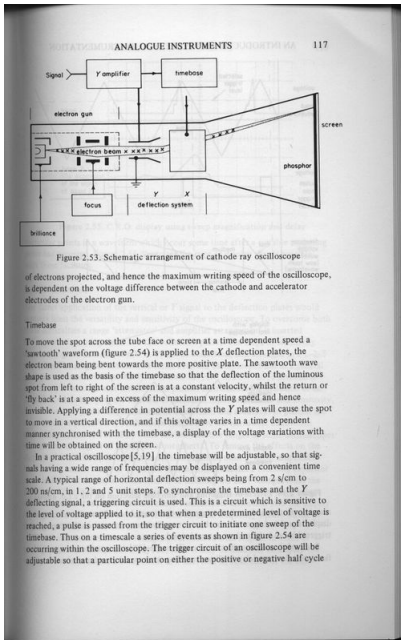
5.4.2. Páginas faltantes

Siempre habrá que ir chequeando que no se nos salteé alguna página, y si ocurre remediarlo acomodando los nombres de los archivos, de forma de que la imagen salteada quede intercalada correctamente. Para esto puede ser útil el `ADVANCED RENAMER`.

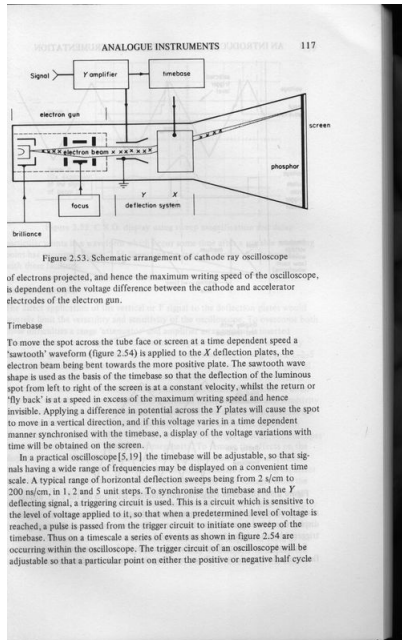
En el ejemplo que estamos trabajando, la página 10 del libro corresponde al archivo `SCAN_022.tif`, es decir que hay un desfase de 12 páginas. Una forma rápida de chequear que no falte ninguna es ver que el archivo `SCAN_112.tif` corresponda con la página 100, el `SCAN_212.tif` con la 200, y así...

De esta forma descubrí que me faltó escanear la página 99 (+ 12 = archivo 111). Usando `ADVANCED RENAMER` desplacé los nombres de los archivos un lugar: `SCAN_112.tif` → `SCAN_113.tif`, `SCAN_111.tif` → `SCAN_112.tif`, ...⁴ De esta forma quedó el nombre `SCAN_111.tif`

⁴ Por si no quedó claro, el programa permite renombrar fácil e instantáneamente todos los archivos de la manera que el usuario elija. No se hizo uno por uno.



(a) Curvatura y sombra.



(b) Contenido fuera del área.

FIGURA 5.4. Distorsiones y sombras sobre el margen interior.

libre para la página que faltaba. Entonces la escané y le puse ese nombre.

En cualquier momento podemos reiniciar el proceso de escaneo a partir del número de página que elijamos.

5.4.3. Curvatura y contenido fuera del área

Si no presionamos bien el lomo sobre el vidrio del escáner, obtendremos imágenes con sombras y distorsiones cerca del margen interior, debido a que el papel no está plano respecto al sensor. Algunos ejemplos en las figuras 5.1 y 5.4.

Si la sombra es pronunciada podría dar problemas durante la binarización. A su vez, si los contenidos están deformados no preservaremos

exactamente el material original y probablemente dificultemos el OCR.

Por otra parte, podría ocurrir que parte del contenido de interés quedase fuera del área de exploración (fig. 5.4).

En todos estos casos, la mejor solución será escanear de nuevo estas páginas.⁵

5.4.4. Distorsiones trapezoidal y por movimiento

La distorsión trapezoidal, llamada en inglés *keystone effect*, es prácticamente inexistente en los escáneres de cama plana, pero se trata de la número uno en los basados en cámaras fotográficas. Es una deformación que se produce por no estar el material perfectamente perpendicular al sensor de la cámara. El ejemplo de la figura 5.5(a) fue tomado con la cámara de un teléfono celular. Si no puede corregirse ajustando el hardware, existen algunas soluciones de software que funcionan si la distorsión es moderada. En particular, SCAN TAILOR tiene la pestaña **Dewarping** en el paso 6, pero aún se considera experimental y no tengo experiencia con ella.

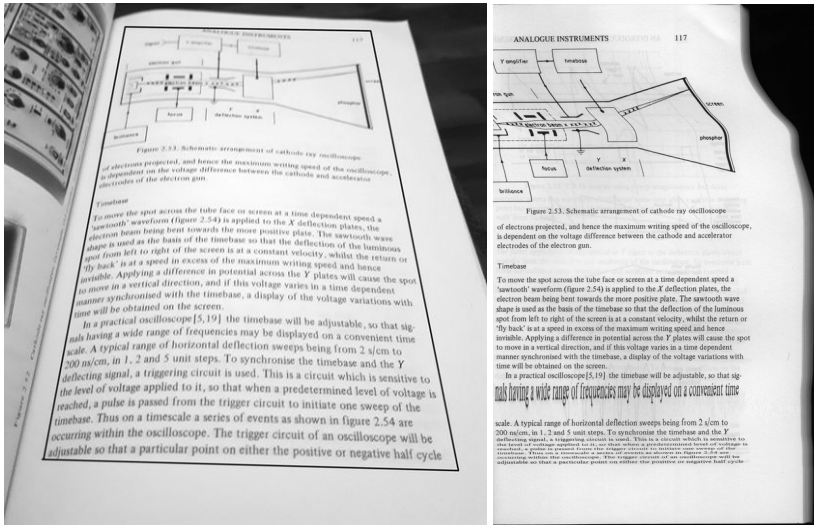
También podría ocurrir que el material se nos moviera mientras pasa el escáner, provocando compresión o estiramiento de las líneas si es en la misma dirección de avance, o deformación lateral si es en dirección perpendicular (fig. 5.5(b)). En este caso también deberá escanearse de nuevo.

5.4.5. Transparencia de páginas

Dependiendo del tipo de papel y la tinta que se hayan usado puede ocurrir que sobre cada página se transparenten los contenidos de las contiguas. Normalmente este es un inconveniente que desaparece durante la binarización, pero en el caso de las fotografías o imágenes complejas es más persistente y en algunos casos es fácil de tratar.

Ampliando en la figura 5.4 vemos que es posible distinguir el texto y los diagramas de las páginas cercanas. En la figura 5.6 se muestra con más detalle.

⁵ Deberemos escanear de nuevo solamente si estas distorsiones afectan al texto o las imágenes. Si apenas caen dentro del blanco de página podremos ignorarlas.



(a) Trapezoidal. Se dibujó el trapecio.

(b) Movimiento.

FIGURA 5.5. Distorsiones trapezoidal y por movimiento.

Para disminuir el efecto de las transparencias sobre las fotografías (fig. 5.7(a)) puede colocarse un papel de color negro⁶ entre la página que se va a escanear y la siguiente (fig. 5.7(c)). Es decir que tendríamos el vidrio del escáner, luego la página que estamos escaneando, luego el papel negro y finalmente las demás hojas. El resultado se ve en la figura 5.7(b).

5.5. Respaldo de los originales

Al terminar es buena idea hacer un respaldo de las imágenes escaneadas, porque a veces las modificaremos directamente, y podremos evitarnos un disgusto si algo sale mal. Generalmente utilizo 7-ZIP para generar un archivo de respaldo con todas. Para hacerlo seleccionamos

⁶ Usé una hoja impresa en negro por las dos caras. Según el caso, podría funcionar mejor otro color.

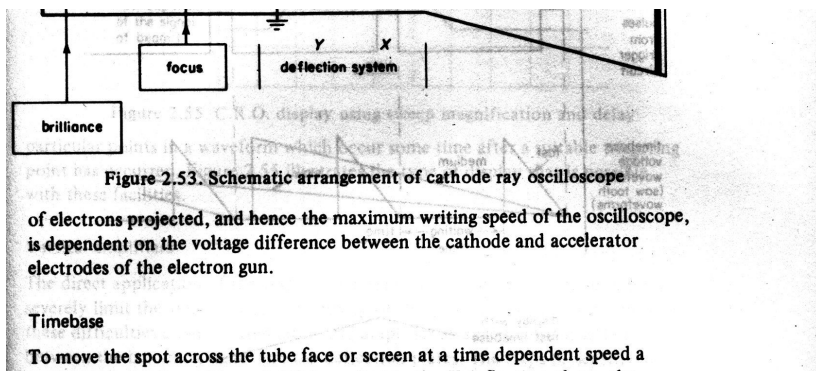


FIGURA 5.6. Transparentación sobre texto y diagramas. (Se ha exagerado un poco para que se distinga con claridad.)

todos los archivos y hacemos  7-Zip >> Add to archive... y elegimos las opciones que deseemos.⁷

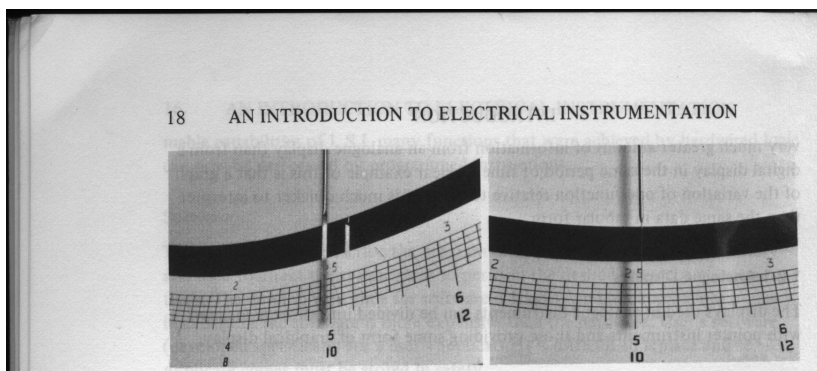
5.6. Conclusiones

En esta sección hemos visto como preparar el escáner y como configurar el software para escanear nuestro material. Se ha usado como ejemplo el software de Epson, pero el aprendizaje debería ser fácilmente trasladable al que tengamos disponible. También vimos como preparar el material a escanear.

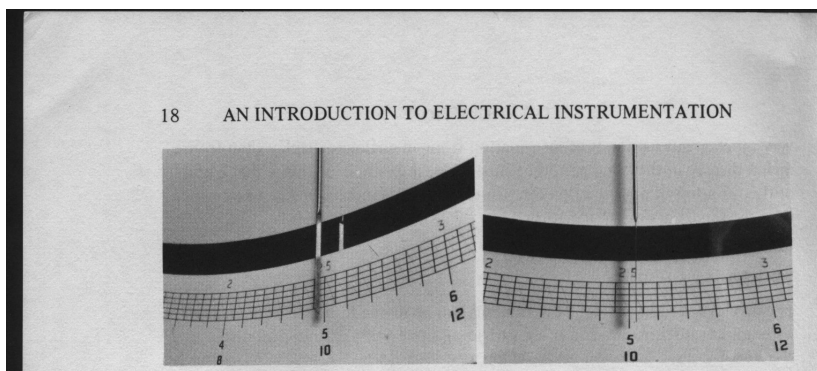
Luego aprendimos sobre el proceso de escaneado en sí, algunos problemas que se pueden presentar y como resolverlos.

En la sección siguiente veremos como procesar los archivos, separando el texto de las fotografías, de forma de obtener imágenes binarizadas aptas para realizar OCR.

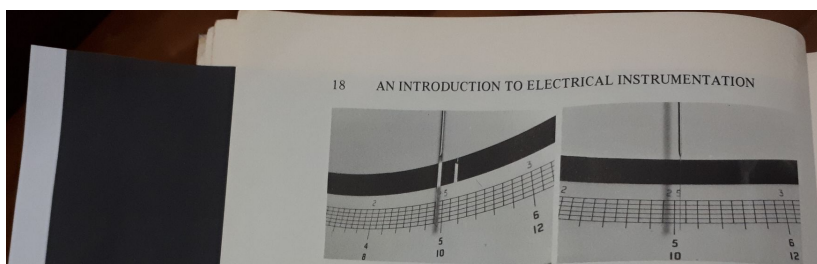
⁷ Las opciones por defecto deberían bastar.



(a) Transparencias sobre fotografías.



(b) Mitigado el efecto usando un papel negro por debajo.



(c) Colocación del papel negro.

FIGURA 5.7. **Detalle de transparencias.**

Capítulo 6

Limpieza y ajuste

Siendo esta la etapa crucial del procesamiento, la presento de la manera más exhaustiva posible. Sin embargo, el detalle que queramos prestarle a cada etapa del procesamiento dependerá de nuestro tiempo, habilidad y paciencia. Algunos de los pasos que siguen podrían ser saltados y aún así obtener resultados aceptables. He tratado de indicarlo cuando así sea.

Lo primero que vamos a hacer es medir las dimensiones del libro usando una regla.¹ En este caso el ejemplar mide 15,5 cm de ancho, 23,5 cm de alto y 2,5 cm de espesor. Al final obtendremos imágenes de 600 dpi, por lo que podemos convertir entre cm y pixeles usando alguna de las siguientes fórmulas para estimar las dimensiones finales de nuestras páginas procesadas:²

$$\text{pix} = \frac{\text{cm} \times \text{dpi}}{2,54} \qquad \text{cm} = \frac{\text{pix} \times 2,54}{\text{dpi}}$$

Resumimos los resultados en el cuadro 6.1.

¹ Si el encuadernado fuera tapa dura (*hardback*) tal vez habría un tamaño para las tapas y otro para las páginas. Será nuestra decisión si preservamos esas diferencias o no.

² En el apéndice B se presentan las correspondientes fórmulas en pulgadas.

(600 dpi)	ancho	alto	espesor
cm	15,5	23,5	2,5
pix	3661	5551	591

CUADRO 6.1. Dimensiones del libro a escanear (en cm-pix).

6.1. Orientación de páginas

De acuerdo a como hayamos colocado el material en el escáner, es posible que las páginas pares o las impares estén rotadas 180°. Si estamos pensando en corregir frecuentemente los errores que vayamos encontrando, como se explica en la sección 6.2.6, es conveniente que la orientación de todas las páginas originales sea la correcta. De lo contrario podemos omitir este paso.

En nuestro caso de ejemplo son las páginas pares las que requieren ajuste. Para hacerlo ejecutamos el script 6.1 de Python que invoca el comando `mogrify` de `IMAGEMAGICK` sobre las páginas de interés. Este nos preguntará por las páginas a corregir (`'Rotate odd or even pages?'`) y el ángulo (`'By which angle?'`) y deberemos ingresar `odd` para las páginas impares o `even` para las pares. Téngase en cuenta lo dicho sobre el respaldo de los originales, puesto que con este script los estaremos modificando.

orientationOddEven.py

```
# This script rotates either odd- or even-named files
↳ from the SCAN directory by the fixed angle.

import os, glob

pages = []
print('Rotate odd or even pages?')
oddEven = input()

if oddEven == 'odd': # odd pages
    pages = [os.path.basename(x) for x in
↳ glob.glob('../SCAN/*[13579].tif')]
```

orientationOddEven.py (cont.)

```

elif oddEven == 'even': # even pages
    pages = [os.path.basename(x) for x in
        ↪ glob.glob('../SCAN/*[02468].tif')]
else:
    print('Wrong option! Try again.')

print('By which angle?')
angle = input() # normally 180 or -180

for x in pages:
    os.system('mogrify -rotate ' + angle + ' ' +
        ↪ x)
    print(x + ' rotated!')

```

SCRIPT 6.1. Rotación de las páginas.

6.2. Procesado con Scan Tailor

Este software es muy completo e incluye una gran cantidad de opciones en sus distintos pasos. La realidad es que al momento de escribir esta guía aún se encuentra en desarrollo, por lo que no existe documentación suficiente que las explique todas en profundidad. Además, algunas son de carácter experimental o directamente no funcionan bien (la función `Dewarp`, por ejemplo). Quien esto escribe no las ha probado todas. Por ello recomiendo al lector que busque la información que necesite en internet³ y que no tenga miedo de experimentar.⁴ Tal vez esta guía esté sugiriendo algún procedimiento manual tedioso para una tarea que ya está cubierta por una nueva opción. La misma recomendación vale para todo otro software que se utilice.

³ El autor del fork original tiene una wiki con información y videos [29].

⁴ SCAN TAILOR no modifica los archivos originales.

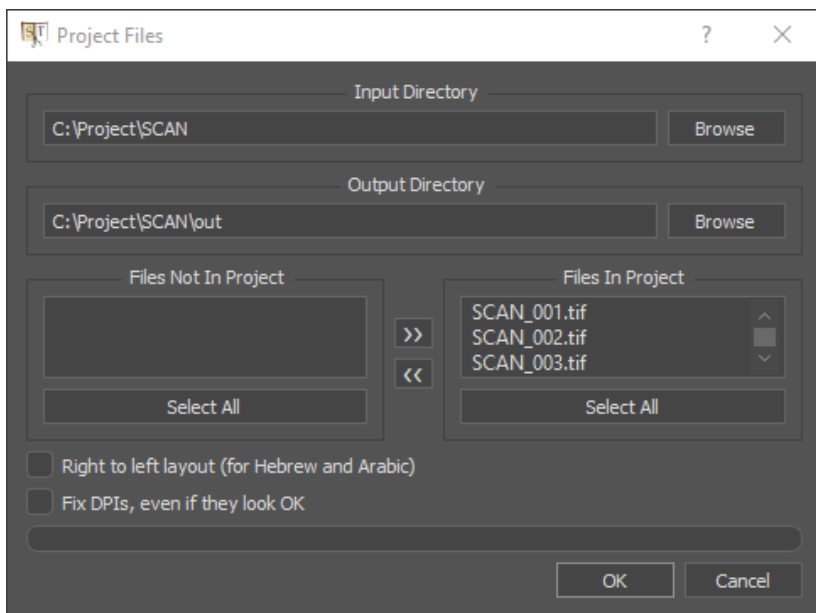


FIGURA 6.1. Cuadro de diálogo **New Project**, de SCAN TAILOR.

6.2.1. Creación de un proyecto nuevo

Dicho esto, comenzamos abriendo el software y seleccionando **File** **New Project**. Aparecerá un cuadro de diálogo como el de la figura 6.1, donde elegiremos la carpeta con las páginas escaneadas. Por defecto las imágenes procesadas irán automáticamente a la subcarpeta **out**. En la columna derecha (**Files In Project**) veremos todos los archivos escaneados. Podríamos excluir algunos seleccionándolos y presionando **<<**, pasándolos a la otra columna (**Files Not In Project**). Como la carpeta **SCAN** era exclusiva para estos archivos, no necesitaremos excluir ninguno.

6.2.2. Paso 1. Fix Orientation

Presionamos **OK** y se cargan las imágenes. La vista debería ser como la de la figura 6.2. Sobre la izquierda tenemos el panel **Filters**,

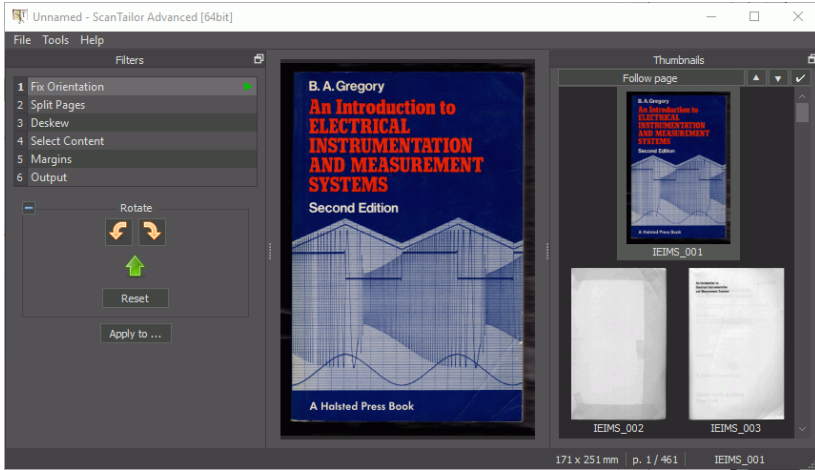


FIGURA 6.2. Ventana principal de SCAN TAILOR. Primer paso: Fix orientation.

con 6 pasos o filtros. Cada uno corresponde a una etapa del proceso. El primero es **Fix Orientation** y se utiliza para corregir la orientación de las imágenes. Si no lo hicimos manualmente como se explica en la subsección 6.1 debemos hacerlo ahora.

Dijimos que teníamos rotadas las páginas pares, por lo que seleccionaremos una cualquiera de estas y presionaremos una de las flechas naranjas debajo de **Rotate** hasta que la página quede en la orientación correcta. Una vez listo presionaremos **Apply to ...** y elegiremos la opción **Every other page**. Para ver que más o menos haya quedado bien, podemos recorrer las miniaturas de la columna derecha y verificar que efectivamente todas las páginas están bien orientadas.

Cuando estemos listos presionaremos ► a la derecha del nombre del filtro. El programa recorre rápidamente todas las imágenes y genera todas las miniaturas. Al terminar podemos ir al paso 2.

Nota: páginas apaisadas Si bien la mayoría de las páginas tienen orientación vertical, a veces nuestro material contiene algunas

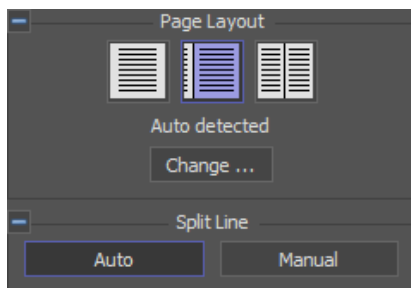


FIGURA 6.3. Segundo paso: Split pages.

páginas con orientación horizontal,⁵ también llamadas páginas «apaisadas» (ver fig. 6.4). Más adelante deberemos decidir como queremos que estén en nuestro documento final (sección 6.2.8). De momento orientémoslas todas tal como aparecen en el material original.

6.2.3. Paso 2. Split Pages

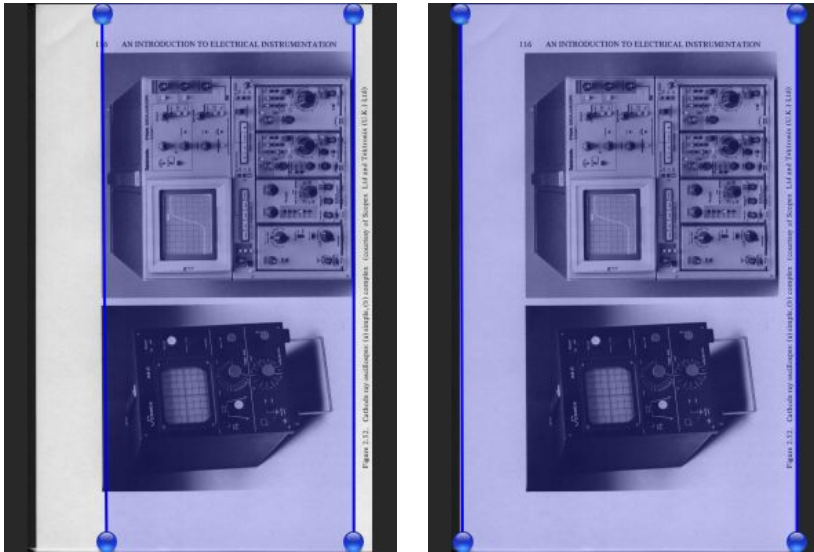
Si hicimos la digitalización de a dos páginas por vez, SCAN TAILOR nos ayuda a separarlas. En **Page Layout** tenemos tres opciones (fig. 6.3). La primera es para páginas que ocupan toda la superficie de la imagen. La segunda permite recortar los lados de la imagen. La tercera es la que efectivamente divide en dos la imagen, separando las páginas.

Cada vez que seleccionamos una página el programa elige la opción que considera más adecuada. También podemos presionar **Change...** y elegir las opciones manualmente para todas las páginas (o pares, impares, etc.).

Presionamos entonces ► para que las analice todas. Luego deberemos recorrerlas y rectificar los errores que encontremos, ya sea cambiando la opción que el programa elige o moviendo la línea por la que se separarán las páginas o recortarán los márgenes. Ver ejemplo en la figura 6.4.

Lo importante en este momento es que todo el contenido quede dentro del área azul y que la máxima cantidad de información inútil quede fuera: trozos de la hoja que no estábamos escaneando, imágenes

⁵ Ver [30] (en inglés).



(a) Error.

(b) Remediado.

FIGURA 6.4. Error en el paso 2: Split pages.

de nuestras manos presionando el material, bandas negras donde no había papel, etc.

6.2.4. Paso 3. Deskew

Deskew se podría traducir por desinclinación. En este paso el programa analiza cada imagen y corrige la inclinación que pueda tener. Si nuestro trabajo de escaneo fue más o menos prolijo entonces cada imagen necesitará rotarse apenas unos grados o menos. En mi experiencia el algoritmo es bastante robusto y se equivoca poco. De todas formas luego de dar ► conviene revisar que esté todo bien.

El programa ofrece la opción `Order by decreasing deviation`, que permite agrupar todas las páginas que debieron rotarse más juntas. Por lo antedicho son las que tienen mayor probabilidad de error. En nuestro ejemplo la mayor desinclinación que se debió aplicar fue de solo 1° . Esto indica que todas las páginas estaban bastante rectas. De todas formas 1°

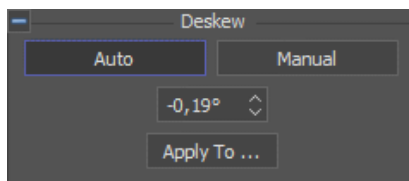
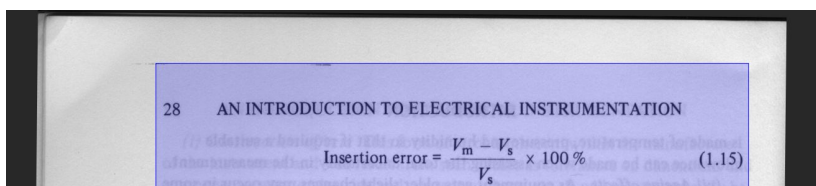
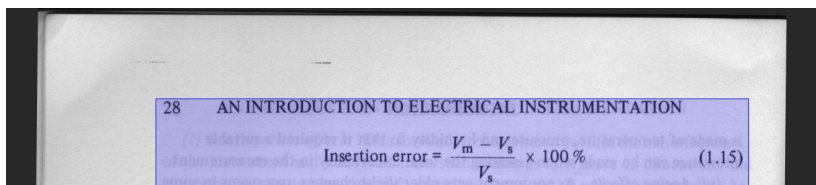


FIGURA 6.5. Tercer paso: Deskew.



(a) Error inducido por defectos en la impresión.



(b) Remediado manualmente.

FIGURA 6.6. Error en el paso 4: Select content.

es una cantidad perceptible a simple vista.

6.2.5. Paso 4. Select content

Esta opción es parecida a la del paso 2, solo que ahora se usará un rectángulo para indicar el área de la página donde está el contenido. Todo lo que quede fuera será descartado del resultado final.

Se trata de un algoritmo relativamente lento y propenso a errores, sobre todo si el material contiene manchas, puntos o defectos similares, que el programa pueda confundir con contenido legítimo (ver fig. 6.6), por lo que recomiendo revisar bien después de dar ►.

Para ayudarnos en esta tarea, también podemos ordenar las páginas: `Order by increasing >> width >> height >> deviation` nos permiten agrupar todas las páginas más altas (bajas), más anchas (angostas) o que se desvían más (menos) de la media juntas. En esos conjuntos es más probable encontrar los errores. Por ejemplo, en la figura 6.6 vemos que la página en cuestión era de las más altas no porque el contenido lo fuera, sino porque no había sido correctamente reconocido.

En otros casos lo será simplemente porque el contenido va más allá de los márgenes usuales del material, común en libros como el nuestro, con muchos dibujos y diagramas; o simplemente porque la ortotipografía del mismo no fue bien cuidada.⁶

Todos los errores de selección de contenido que no corriamos seguramente se hagan evidentes al compaginar el documento, sobre todo si somos escrupulosos con los márgenes. En ese punto serán aún fáciles de corregir, pero cuanto menos nos quede hacia adelante, mejor.

6.2.6. Paso 6. Output

¿Y el paso 5? Bueno, hay una razón por la que recomiendo ir primero al paso 6: muchas de las correcciones que realizaremos en este paso afectarán los ajustes del paso 5. Por ello, será mejor no pasar al 5 hasta que hayamos terminado con el paso 6.

Comenzaremos por seleccionar `Output Resolution (DPI) >> 600` y presionamos ► para que se binaricen todos los archivos. Este es un proceso que también le lleva a SCAN TAILOR algo de tiempo, aunque se ha mejorado mucho respecto a la versión original. Al finalizar podemos inspeccionar cada página en busca de errores. A continuación señalamos algunos de los más comunes.

Caracteres mal impresos o faltantes

A veces el proceso de impresión falla y algunos caracteres aparecen cortados, como en la figura 6.7. Existen dos formas de corregirlos.

1. Usando GIMP para editar los originales. Tenemos la herramienta pincel para corregir errores o podemos copiar caracteres buenos y pegarlos sobre los que tienen problemas. Es bastante versátil

⁶ Como en nuestro material, donde el texto no está justificado.

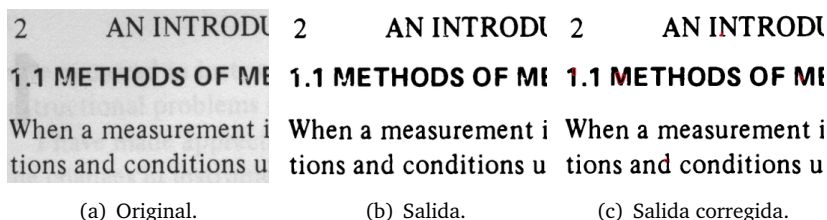


FIGURA 6.7. **Caracteres cortados.** La corrección se hizo desde la pestaña **Fill zones**.

pero requiere buscar la imagen, cargarla, editarla, guardarla, ver el resultado en SCAN TAILOR y quizás repetir.

2. Usando la pestaña **Fill zones** de SCAN TAILOR. Allí podemos dibujar regiones (poligonales o libres) para que el programa rellene. Haciendo  **Pick color** sobre ellas elegimos el color para rellenar (generalmente blanco o negro).

Dependiendo de la cantidad y gravedad de estos errores valdrá la pena invertir el tiempo y repararlos o simplemente dejarlos así. Si los originales tienen problemas hay un límite en la cantidad de esfuerzo que podemos echarle para mejorarlos. No hay que ser más papista que el papa.

Motas y manchas no detectadas

Una de las opciones que tiene al etapa 6 se llama *despeckle*, que podría traducirse como eliminación de motas. Lo que hace es buscar puntos pequeños y generalmente aislados, que probablemente no sean parte del contenido, sino más bien problemas de impresión o escaneo (motas de polvo).

Funciona bastante bien, salvo en dos casos: si los puntos son demasiado grandes (fig. 6.8) o si están entremezclados con el resto del contenido (fig. 6.9). En ambos se puede corregir manualmente, ya sea usando GIMP o **Fill zones**.

También existe la posibilidad de cambiar el parámetro del filtro, para que detecte detalles más grandes. En dicho caso se debe tener cuidado

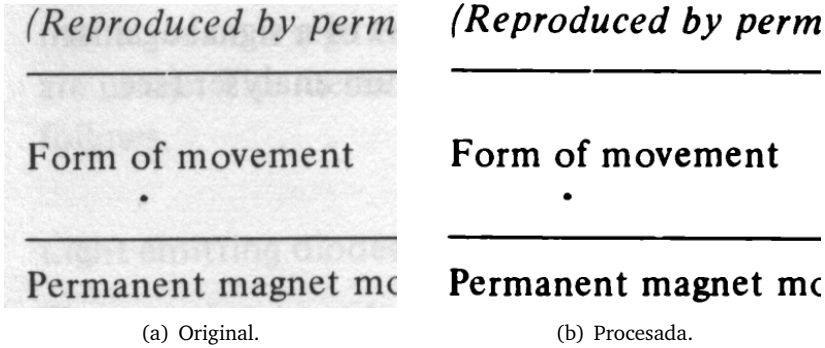


FIGURA 6.8. Filtro *despeckle* frente a puntos demasiado grandes.

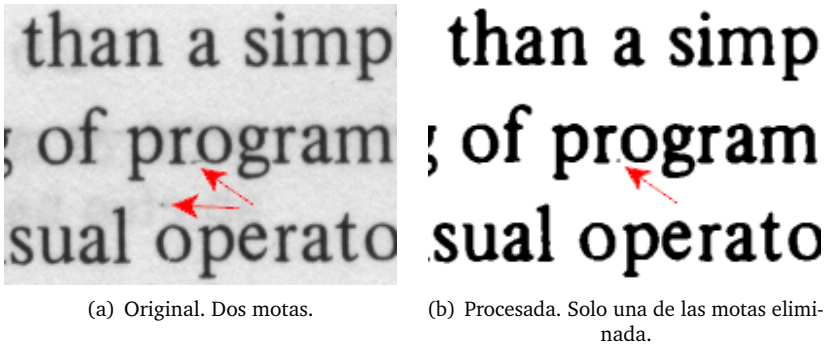
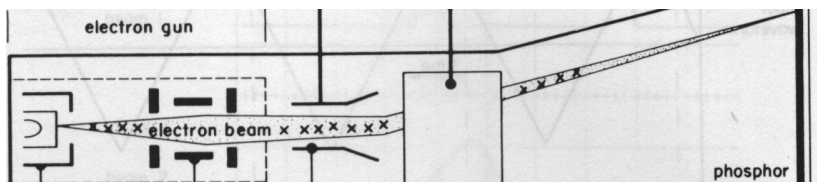


FIGURA 6.9. Filtro *despeckle* frente a motas pegadas al texto.

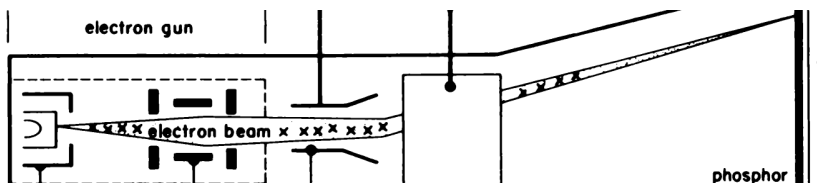
de que no elimine elementos que no debe, como por ejemplo puntos y comas.

Contenido por motas

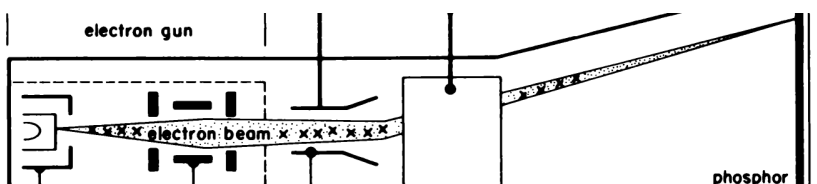
A veces al *despeckle* se le va la mano y borra puntos que en realidad pertenecen al contenido. Ver figura 6.10. Si interesa retener esos puntos podría desactivarse el filtro y hacer un *despeckle* manual en esa página. Otra alternativa puede ser darle énfasis a los puntos de interés con el GIMP, para que el filtro no los borre (es lo que se hizo en este caso).



(a) Original.



(b) Procesada.



(c) Procesada luego de dar énfasis con GIMP.

FIGURA 6.10. Efecto indeseado del filtro despeckle.

Desinclinación incorrecta

Si bien este error debió comentarse en la sección 6.2.4, al menos a mí me sucede que no lo detecto hasta este paso.

Se trata de páginas que no se rotaron el ángulo correcto. En mi experiencia, sucede principalmente en dos casos:

1. Páginas apaisadas. Nuestro libro tiene varias tablas grandes y páginas con figuras que se imprimieron rotadas para que entren mejor en la hoja. Debemos ajustar el ángulo manualmente.⁷
2. Páginas con contenido a distintos ángulos. En nuestro libro sucede que algunos dibujos consisten de líneas rectas no paralelas al texto.

⁷ Téngase en cuenta lo que se explica en la sección 6.2.8.

Esto puede confundir al algoritmo. En otros casos podría haber texto escrito a distintos ángulos, ya sea de manera intencional o por errores en la composición de la página. Dependiendo del caso, podemos ajustar el ángulo manualmente o usar GIMP para alinear solo aquella parte de la página que no lo esté.

Picture zones

Para las páginas que además del texto tengan imágenes a color o en escala de grises que queramos preservar, deberemos elegir la opción **Mode** >> **Mixed**.⁸ La opción por defecto es **Black and White**, que binariza todo, y la restante, **Color / Grayscale**, retiene la página original completa, salvo por los efectos de los pasos anteriores.

Elijiendo **Mixed** el software intentará adivinar las áreas donde se encuentran las imágenes. En la pestaña **Picture Zones** se señalan titilando con color azul. Por defecto estas áreas pueden tener cualquier forma, pero en el submenú **Picture Shape** >> **Rectangular** (fig 6.11) podemos forzar a que sea rectangular, que servirá en la mayoría de los casos.

En los que no, deberemos dibujar nuestras propias zonas (y tal vez eliminar o corregir algunas mal detectadas). Para cada una podemos hacer  >> **Properties** y elegir si añadirla a la capa ya detectada (**Add to auto layer**), eliminarla de esa capa (**Subtract from auto layer**) o eliminarla de todas las capas (**Subtract from all layers**). Todo esto tendrá sentido si las zonas se superponen. Sugiero experimentar un poco con las opciones para entender bien su significado.

Un ejemplo de zonas dibujadas manualmente y su resultado se muestra en la figura 6.12. Nótese que es posible tener huecos dentro de la imagen que serán rellenados de blanco. En este caso le darán un efecto interesante al resultado final. De lo contrario esos huecos quedarían rellenos con el color del blanco de la página, que como se ve en la figura 6.12(a) es un beige.

Activar **Options** >> **Equalize illumination (Color)** me ha dado buenos resultados. Téngase en cuenta.

⁸ Este es el método que recomiendo para trabajar con las página con fotografías de nuestro material. Algunas alternativas no recomendadas se discuten en el apéndice D.

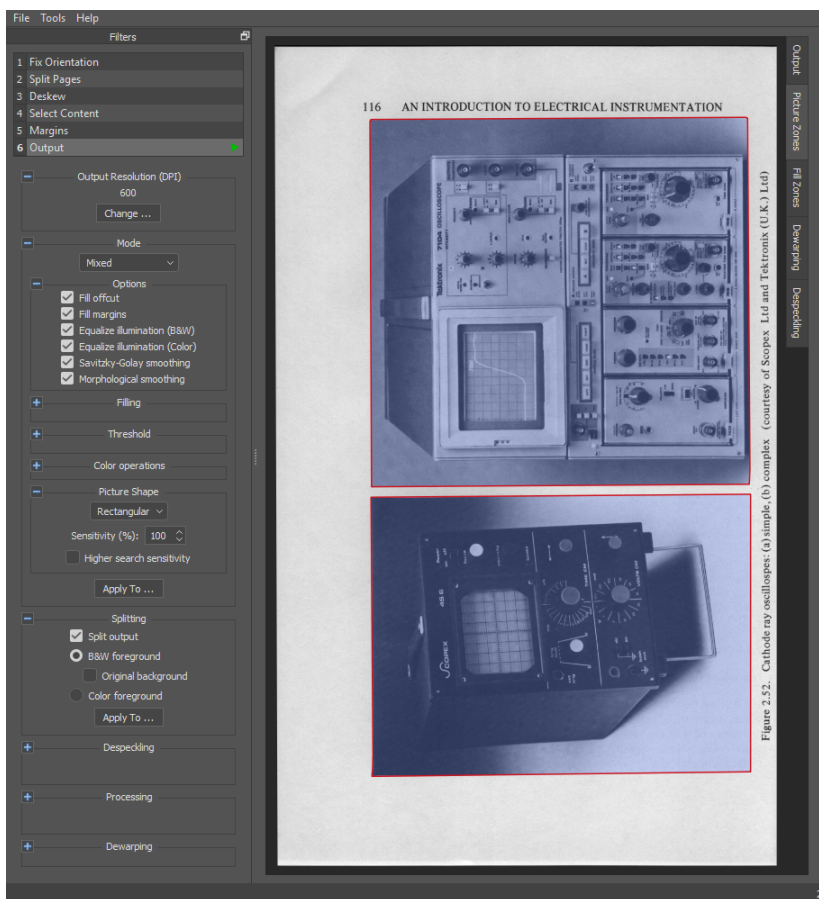
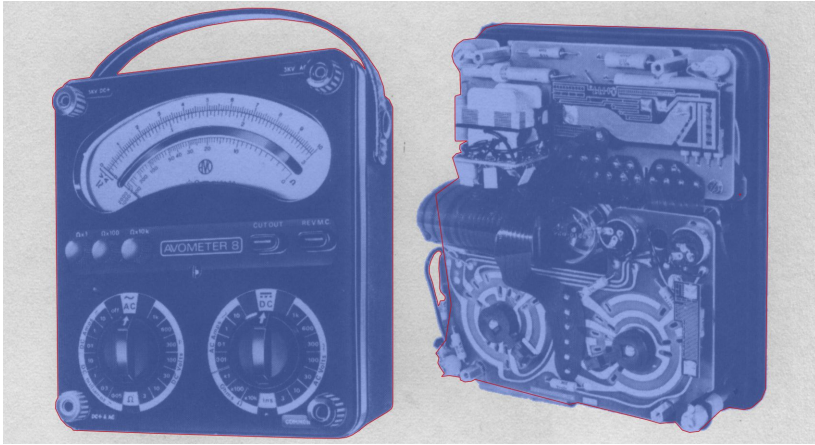
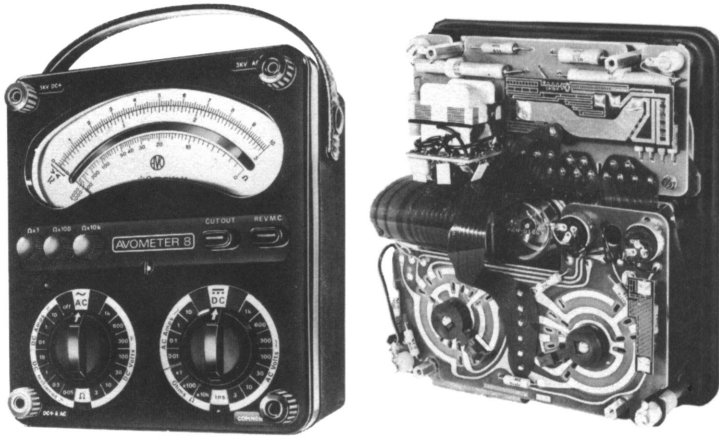


FIGURA 6.11. Sexto paso: Output. Se seleccionó la pestaña **Picture zones**.



(a) Zonas marcadas manualmente.



(b) Resultado.

FIGURA 6.12. Selección manual de la imagen en pestaña **Picture Zones**. Los contornos rojos se dibujaron manualmente.

Para estas páginas sugiero activar siempre **Splitting** **Split output** ☒ **B&W foreground**. Esto generará dos carpetas más dentro de **out**: **foreground** y **background**, adonde se guardará cada componente de la imagen. A la primera irá el todo el contenido binarizado y a la segunda las fotos. Podemos activarla y echar un vistazo para entenderlo mejor.

Esto nos permitirá procesar cada parte en sus particularidades de manera independiente, de forma de lograr un resultado óptimo. Si no se va a hacer, de todas formas elegir la opción no hace daño, pues en **out** se sigue guardando la imagen con las dos componentes.

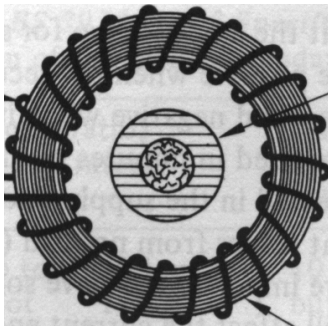
Dibujos problemáticos

Algunos dibujos o diagramas, a pesar de estar impresos a una sola tinta, pierden muchos detalles cuando son binarizados. Ocurre cuando hay líneas muy cercanas pero que no se tocan: el blanco que queda en medio no es lo suficientemente claro, y SCAN TAILOR lo convierte en negro. Análogamente si hay líneas negras demasiado tenues. La figura 6.13 ilustra un ejemplo. Para arreglar esto existen tres opciones.

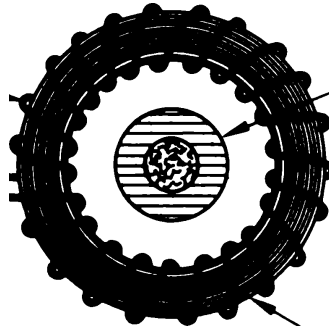
1. La más sencilla es tratarlos como imágenes. Procedemos exactamente como se explicó antes.
2. Si se dispone de un poco más de tiempo, se pueden editar en el GIMP para generar un contraste mayor que permita una binarización más efectiva. Para ello se pueden usar filtros o manualmente redibujar los detalles que se quieran conservar.
3. La tercera consiste en ajustar el algoritmo de binarización, como se explica a continuación. Tiene el inconveniente de que los cambios afectan a toda la página y no solo al dibujo de interés.

Algoritmo de binarización

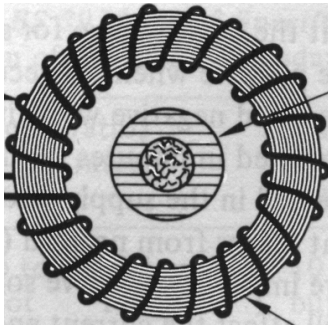
Existen diferentes algoritmos de binarización de imágenes, cada uno con ventajas y desventajas. Con el submenú **Threshold** (umbral) SCAN TAILOR permite cambiar entre **Otsu** (por defecto), **Sauvola** y **Wolf**. De no estar conformes con el resultado por defecto, podríamos intentar cambiando el algoritmo o ajustando los parámetros que se ofrecen. El



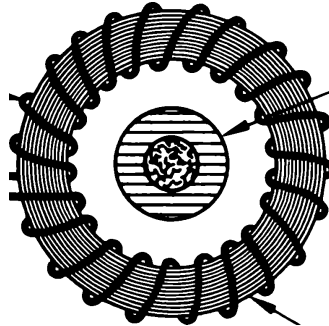
(a) Escan original.



(b) Salida original.



(c) Corregido (manualmente).



(d) Salida final.

FIGURA 6.13. Binarización de un dibujo problemático.

impacto que tiene cada uno está por fuera del alcance de nuestra guía, si bien existe abundante literatura al respecto.

6.2.7. Paso 5. Margins

Una vez que revisamos cada página y acomodamos bien los contenidos, eliminamos manchas espurias, etc., es hora de ajustar los márgenes. Propongo un método que conserva la apariencia original y otro que introduce márgenes arbitrarios.

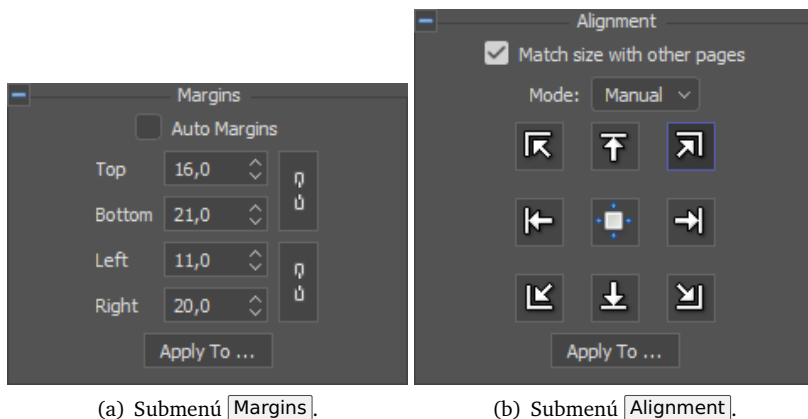


FIGURA 6.14. Quinto paso: Márgenes.

Conservando la apariencia original

Si estamos buscando reproducir las dimensiones del material original deberemos tener los cuidados que se explican a continuación.

Para ello comenzamos decidiendo respecto a qué parte de las páginas vamos a referir los márgenes. En el caso que nos ocupa, el libro tiene los números de páginas impresos en las esquinas superiores exteriores de cada hoja. Las pares en la superior izquierda y las impares en la superior derecha. Estos son referencias bastante naturales y son las que usaremos.

Vamos entonces al paso 5 de SCAN TAILOR, nos paramos en una página par y en el submenú **Alignment** (fig. 6.14(b)) presionamos la flecha que apunta hacia arriba y a la izquierda. Veremos que la alineación cambia. Por ahora no nos preocuparemos por el ancho de los márgenes. El paso siguiente es presionar **Apply To...**, seleccionar **Every other page** y dar **OK**. Luego hacemos lo mismo con las páginas impares, eligiendo la flecha que apunta arriba a la derecha.

Si nuestra alineación fuera por ejemplo sobre los números de página que aparecen abajo al centro u otra, deberemos elegir la flecha que mejor se ajuste al resultado buscado.

Para las tapas probablemente no usemos márgenes, por lo que des-

página	nro.	sup.	inf.	izq.	der.
alta	p. 296	16	21	—	—
ancha (par)	p. 250	—	—	20	11
ancha (impar)	p. 117	—	—	11	20

CUADRO 6.2. Márgenes ingresados para las distintas páginas (en mm).

activamos ☐ Match size with other pages. Veremos que al hacerlo las dimensiones finales de estas páginas no afectarán a todas las demás. Podría ser necesario activar esta opción también para aquellas páginas que por alguna razón tengan un tamaño físico mayor o menor que el resto, como puede ocurrir con libros que traen páginas dobladas que deben extenderse.

Como bien sabemos, hay muchas páginas dentro del mismo libro que no siguen el mismo patrón, a estas las llamaremos páginas atípicas. El conjunto de atípicas se compone generalmente de páginas con grandes imágenes o diagramas, tablas, o aquellas que dan comienzo a un capítulo o sección. El ajuste de estas lo dejaremos para el final.

De momento concentrémonos en las típicas. Elijamos ordenar las páginas por alto () y busquemos abajo del todo la típica más alta. En nuestro caso al final aparecen las tapas (obviamente) y páginas con tablas. La típica más alta es la p. 296. Vamos entonces a esa página en el libro y medimos los márgenes superior e inferior con la regla. Ingresamos esos valores y presionamos .

Ahora pasamos a ordenar por ancho: . Haremos lo mismo que antes, buscaremos la típica más ancha par e impar. Son las pp. 250 y 117. Medimos para cada una el margen izquierdo y derecho, los ingresamos y elegimos , para cada una.⁹

Las medidas en este caso se presentan en el cuadro 6.2.

⁹ Las medidas dieron 20 mm y 11 mm para las pares, y 10 mm y 19 mm para las impares. Debido a tolerancias de impresión y/o encuadernado las páginas no son simétricas, por lo que las medidas de los márgenes tampoco lo serán exactamente. Para no complicarnos tanto haremos que en nuestro resultado sí lo sean, simetrizando estos valores.

SCAN TAILOR por defecto impone el mismo valor para los márgenes izquierdo-derecho y superior-inferior. Si, como en este caso, queremos ingresar distintos anchos para cada margen, debemos presionar  a la derecha en la figura 6.14(a). Así, la cadena se abre  y los anchos pueden ser arbitrarios.

Por haber cambiado los márgenes y la alineación deberemos regenerar todos los archivos de salida. Pero por el momento elijamos una página cualquiera y vayamos al paso 6. Buscando en la carpeta  out podemos ver de que dimensiones quedó. En este primer intento obtuve un archivo de 3955 pix por 5790 pix. Aplicando la fórmula que se da al principio de la sección concluimos que medirá 16,7 cm por 24,5 cm, lo que en ambas dimensiones es un poco más grande que el original físico. Esto seguramente se deba a las páginas atípicas, para las que deberemos ajustar manualmente los márgenes.

Volvemos al paso 5 y ordenamos por altura. Vamos a buscar los *outliers* que nos están agrandando todas las páginas. Les medimos los márgenes e ingresamos esos datos. Lo mismo ordenando por ancho.

Al terminar, deberíamos tener archivos de dimensiones adecuadas. En este caso obtuve 3663 pix por 5554 pix, lo que es menos del 0,1 % de diferencia con lo que esperábamos. Yo lo doy por bueno.

Alineación Aunque los tamaños ya están bien, aún falta alinear las páginas. Podríamos dejarlo así, pero quedaría mejor si todos los números de página ocuparan la misma posición en todas las páginas, independientemente de que haya contenido que invada los márgenes. Para hacer este ajuste, SCAN TAILOR Advanced ofrece las así llamadas *guides* (fig. 6.15), que pueden ser verticales u horizontales.

Para crearlas hacemos  >> Add vertical guide. Haciendo  +  + [arrastrar] podemos mover el contenido en sentido horizontal para alinearlo. En este caso será al número de página. Necesitaremos una para las pares y otra para las impares. Sugiero fijar las guías en las páginas más anchas (pp. 250 y 117).

Con  +  sobre el área que queremos alinear, SCAN TAILOR hace un buen trabajo de acomodar automáticamente el contenido a la guía más cercana. Si no nos satisface el resultado, podemos arrastrarlo manualmente como en el párrafo anterior, hasta colocarlo donde queramos.

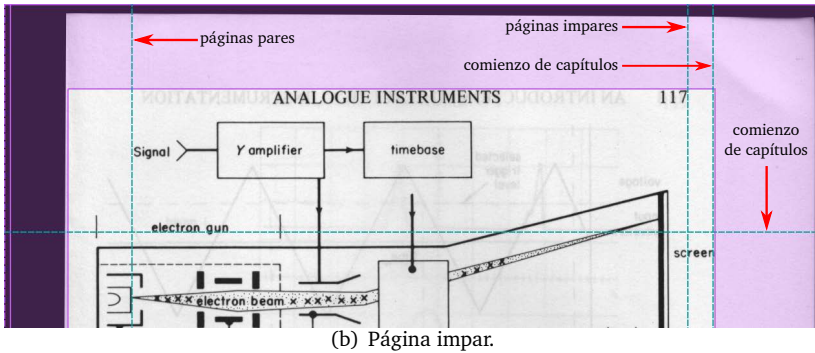
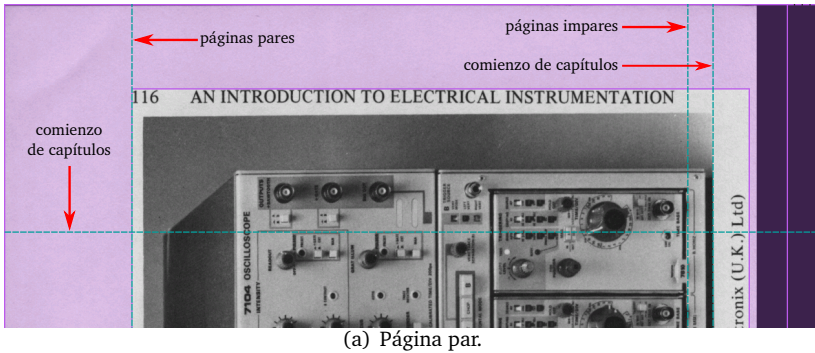


FIGURA 6.15. Detalle de las guías para ajustar márgenes. Las páginas (típicas) ya fueron alineadas con sus guías correspondientes.

Si también quisiéramos ajustar la posición vertical, crearíamos una guía horizontal con  Add horizontal guide y usaríamos  +  o  +  + arrastrar para alinear el contenido verticalmente. Podría ponerse esta guía en los títulos de página, en cuyo caso la crearíamos con la página más alta como referencia. En este caso considero que no es necesario ajustar verticalmente las páginas típicas.

Como todos los comienzos de capítulos omiten el número de página, debemos usar otro punto como referencia. Elegí la base del texto que marca el título. Es la única guía horizontal que creé para este proyecto.

Algunas otras páginas atípicas pueden alinearse sin la necesidad

de guías. Bastará con medir los márgenes e ingresarlos manualmente. Estoy pensando en las portadillas, portadas, colofones, etc.

Para sacar provecho de este método debemos avanzar manualmente página por página, alineando el contenido a las guías. Este puede tomar algo de tiempo y según el caso podría no valer la pena.

Verificación Si nos interesa verificar que los márgenes nos hayan quedado bien, podemos usar DJVU SMALL con un perfil cualquiera (por ejemplo, `default`) para generar un DJVU de prueba y, pasando las páginas rápidamente, detectar algún error de alineación que no hayamos notado.¹⁰ También es posible ir pasando dentro de SCAN TAILOR, pero de momento la carga es más lenta y a veces no permite apreciar las diferencias.

Alternativa

Si mantenerse fiel al material original no es el objetivo, se pueden elegir márgenes razonables, tal vez discriminando un poco para páginas pares e impares (lo mismo respecto de la alineación) y aplicar masivamente a todas las páginas, editando manualmente alguna si así se desea. El resultado dependerá enteramente de nuestro gusto y parecer, pero recomiendo usar los márgenes aproximados del material original al menos como referencia.

6.2.8. Páginas apaisadas

Recomiendo corregir la rotación de las páginas apaisadas (*landscape*, en inglés) para que estén bien orientadas en el documento final (ver figura 6.16). De esta forma se facilita el OCR y queda más fácil de visualizar. Lo más correcto sería elegir la orientación adecuada desde el paso 1 de SCAN TAILOR. No lo hacemos así porque no tener todas las páginas orientadas en el mismo sentido nos dificulta la configuración de márgenes consistentes.¹¹ Por ello, debemos tomar nota de los números

¹⁰ Para aprender a trabajar con DJVU SMALL consultar la sección 9.2.

¹¹ En realidad, nos obliga a considerar todas esas páginas como atípicas, teniendo que ingresar todos sus márgenes manualmente.

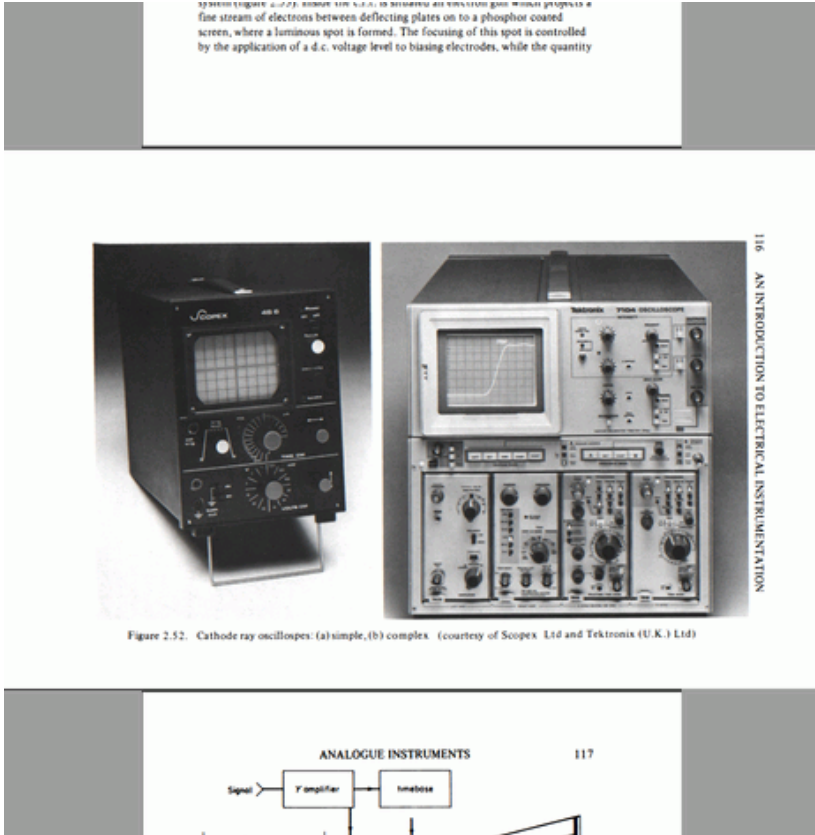


FIGURA 6.16. Página apaisada y sus contiguas en el documento final.

de página que queramos ajustar y adaptar el script 6.2 para que las ajuste correctamente.¹²

¹² Bastará con rellenar la variable `landscapePages` con los números de página correspondientes.

fixLandscapePages.py

```
# This script rotates all the landscapePages by the  
↪ fixed angle.  
  
import os  
  
landscapePages = [128, 161, 210, 213, 214, 224, 377,  
↪ 379, 381, 382, 384, 385, 386, 387, 388, 407, 435,  
↪ 438, 439, 440]  
angle = '180'  
  
for x in landscapePages:  
    os.system('mogrify -rotate ' + angle + '  
↪ ../SCAN/out/SCAN_' + str(x) + '.tif')  
    print('SCAN_' + str(x) + '.tif' + '  
↪ rotated!')
```

SCRIPT 6.2. Ajuste de páginas apaisadas.

6.3. Tapas

Las tapas de los libros muchas veces son las únicas ilustraciones a color que que estos incluyen, por lo que me gusta dedicarles un poco más de atención.

En la figura 6.17(a) vemos la tapa del libro sobre el que estamos trabajando. Consta básicamente de 4 colores: blanco para el nombre del autor y la edición, naranja para el título de la obra, azul oscuro para el fondo y los gráficos de la parte de abajo y azul celeste para el fondo del gráfico. Podemos ver que en las esquinas presenta marcas de doblez y que en algunos lugares se ha borrado la ilustración.

Para remediar estos detalles uso GIMP. El proceso cae un poco por fuera de los objetivos de esta guía, así que no entraré en pormenores. Valga decir que selecciono las áreas de cada color y las vuelvo uniformes, para luego convertir la imagen en indizada. De esta manera conservo el diseño original y reduzco el tamaño casi 100 veces. La diferencia se muestra en la figura 6.17.

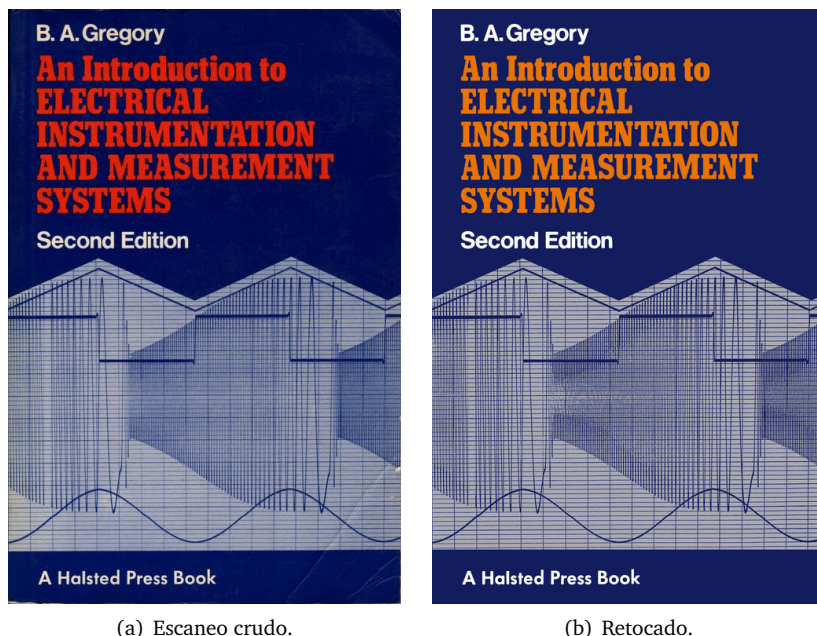


FIGURA 6.17. Efectos de retoque en la tapa del libro. Se pasó de un TIFF de 56 MB a uno de 686 kB con similar apariencia.

Las cubiertas con diseños más complejos o que incluyen imágenes no admiten el mismo tipo de edición, aunque muchas veces es posible mejorarlas ajustando brillo, contraste, etc.

6.3.1. Nota: sobre colores

No tiene mucho sentido obsesionarse con que los colores se vean exactamente igual que en el material físico.

Tenga en cuenta que si su monitor no está calibrado, los colores que está viendo en él no se corresponderán con los que se verán por ejemplo si mandara a imprimir lo que tiene en frente. No olvide que estamos digitalizando un documento, por lo que estamos pensando en visualizarlo luego en una pantalla. Distintas personas verán distintos colores en distintos monitores. No hay remedio para esto.

Ocr con FineReader

Una vez que las páginas fueron limpiadas y ajustadas y las imágenes separadas, podemos proceder a realizar el reconocimiento de texto. Como paso previo, notemos que si marcamos **Picture zones** y elegimos la opción **Split output**, tenemos en la carpeta **out** las imágenes procesadas y dentro de la misma otras dos carpetas, como se explicó antes.

Debemos ahora copiar las imágenes desde **foreground** a la carpeta **out** y sobrescribir los archivos. Así, en **out** habrá solo imágenes binarias con el texto. Las fotos quedarán en **background** y trabajaremos sobre ellas más adelante.

Lo mismo haremos con las tapas si las procesamos aparte. Las copiamos a la carpeta **out**, sobrescribiendo las que generó SCAN TAILOR. De esta manera tendremos en **out** las páginas que conformarán el documento final (salvo las fotos).

Nota previa FINEREADER también es un programa muy completo, con una cantidad de modos de funcionamiento, opciones y errores que esta guía no pretende cubrir exhaustivamente. Si alguna explicación no quedara clara, lo mejor será experimentar, leer el manual de usuario o realizar consultas en algún sitio especializado.

7.1. Configuración

Abrimos luego el FINEREADER¹ y procedemos a configurar primero las opciones básicas (más adelante trataremos algunas otras). Se accede al menú haciendo  **Tools** **Options**. En principio sugiero configurar las opciones tal cual se muestran en las imágenes.

7.1.1. Pestaña Image Processing

En la pestaña **Image Processing** presionamos **Show Advanced Settings** y deshabilitamos todas las opciones, como en la fig. 7.1. Así nos aseguramos de que el software no realizará ninguna corrección de imagen. Para eso ya transitamos por los pasos anteriores. Si aún hubiera algo para ajustar, lo mejor será volver a SCAN TAILOR y hacerlo nosotros mismos.

7.1.2. Pestaña Languages

Pasamos ahora a la pestaña **Languages** (fig. 7.2) donde crearemos primero un nuevo idioma basado en uno ya existente. Esto lo hacemos así si nuestro material contiene caracteres de otro alfabeto, por ejemplo el griego. O si queremos editar los caracteres permitidos para ayudar con el reconocimiento.²

Elegiremos ☒ **Specify OCR languages manually**. Luego presionaremos **New** y seleccionaremos ☒ **Create a new language based on...** (fig. 7.3(a)). En este caso elegimos **English**, presionamos **OK**, le ponemos por nombre **MyEnglish** y damos **OK** nuevamente (fig. 7.3(b)). Al final deberá decir ☒ **Specify OCR languages manually** **MyEnglish**; (fig. 7.2).

¹ Recomiendo trabajar con la versión 14 (la más reciente al momento de escribir esta guía). Sin embargo, cualquier versión que permita salvar en DJVU es suficiente (v11 o superior). Las opciones no serán exactamente iguales (habrá que improvisar) y no podremos usar el script 7.1 (no es grave).

² A veces el programa confunde un carácter por otro. Muy comúnmente «l» cursiva por «/». Si este último no es usado en nuestro material, podemos eliminarlo de la lista de caracteres permitidos para forzar al programa a intentar reconocer otro carácter. Esto no garantiza resultados (podría aun confundirlo con «|», por ejemplo), pero puede ayudar.

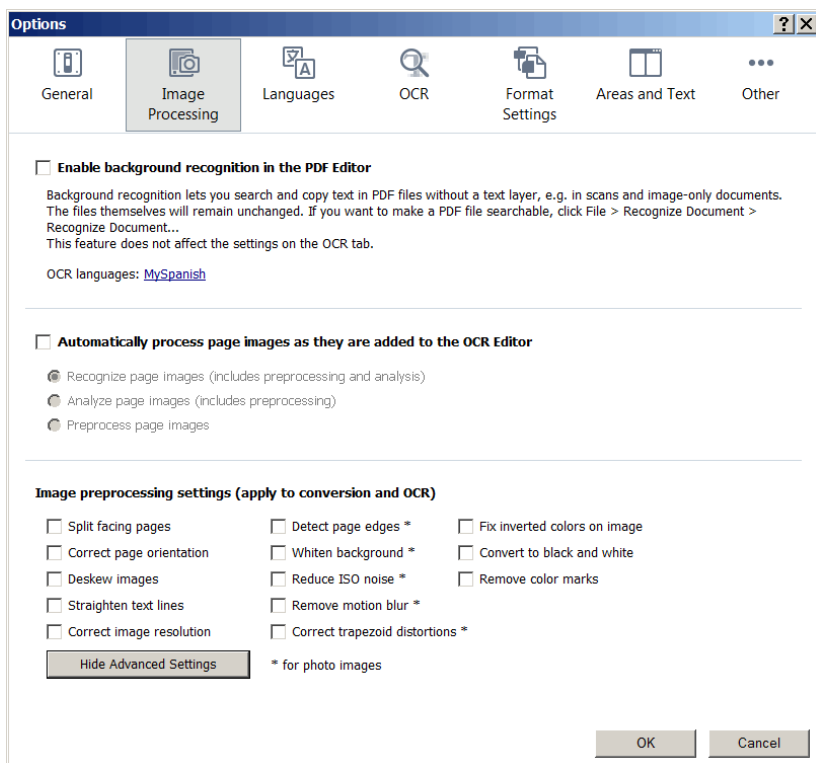


FIGURA 7.1. Opciones de FINEREADER. Pestaña **Image Processing**.

7.1.3. Pestaña Format

Finalmente, en la pestaña **Format Settings** y elegimos el formato **PDF**. Allí configuramos las opciones como en la figura 7.4(a). Es importante que en el menú desplegable **Image quality** elijamos la opción **Custom...** y que desmarquemos ☐ **Reduce original resolution if required**, elijamos **Color control** > **Do not change image color** y también **Quality** > **Quality loss not allowed**. Además, en **Searchable PDF settings** elegimos

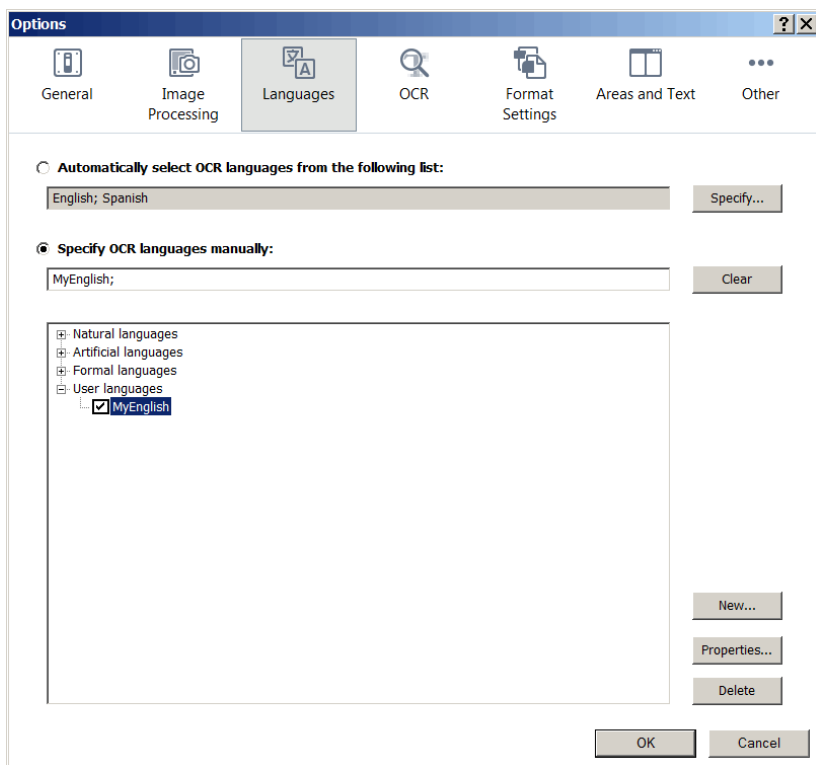


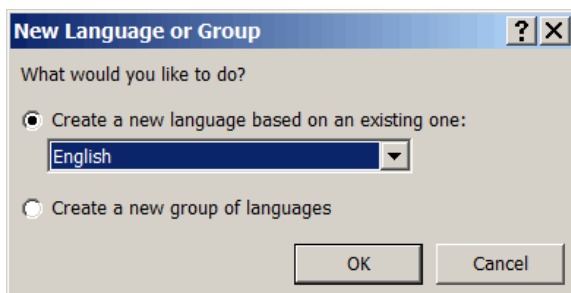
FIGURA 7.2. Opciones de FINEREADER. Pestaña **Languages**.

Text under the page image, ☐ Create bookmarks from headings,³ marcamos ☒ Create PDF tags y seleccionamos Use Windows fonts.⁴

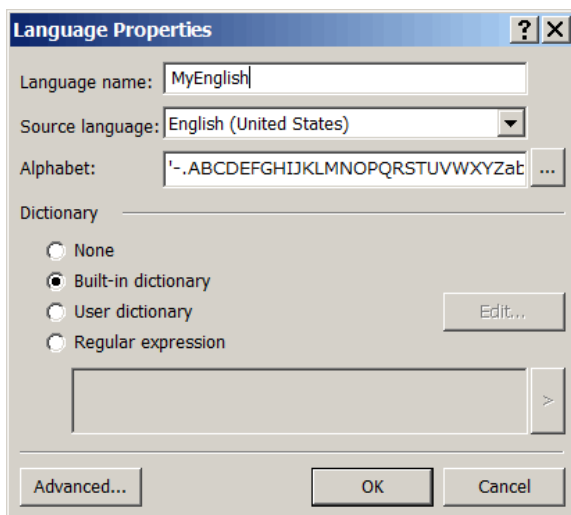
De forma similar, configuramos las opciones del formato DJVU como en la figura 7.5. Aquí la única opción imprescindible será **Save mode** > **Text under page image**, puesto que el DJVU que genera FINEREADER no lo usaremos más que para «robar» el texto reconocido e insertarlo

³ La he probado y no da buenos resultados. Además, crearemos nuestros propios marcadores más adelante.

⁴ No tengo justificación para esta opción. No sé que consecuencias pueda tener desde el punto de vista legal. Creo que la visualización correcta queda a salvo con el PDF/A, que veremos más adelante.



(a) Creando un nuevo idioma.



(b) Propiedades del idioma.

FIGURA 7.3. Opciones de FINEREADER. Creando un nuevo idioma.

en otro archivo DJVU, que vamos a generar más adelante con otros programas.

7.2. Creación del proyecto

Luego presionamos **File** > **New OCR Project** e inmediatamente **File** > **Save OCR Project**. Vamos a guardar el proyecto en **Project** ▶ **ABBY**. De esta manera, FINEREADER irá guardando los cambios a medida que vayamos procesando las páginas.

Luego, cargamos las páginas haciendo **File** > **Open Image**, seleccionando todos los archivos de la carpeta **out** y presionando **Open**. Tomará un tiempo para cargarlas todas y luego la ventana quedará como en la figura 7.6.

7.3. Reconocimiento

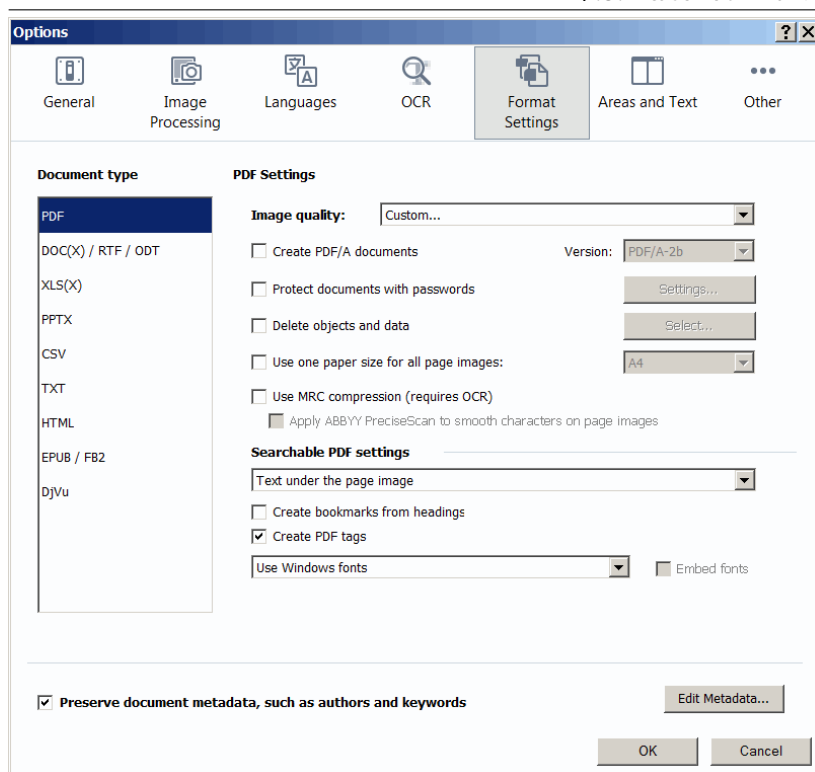
Esta es la parte sustanciosa de toda la etapa de OCR.

Como siempre, dependerá de nosotros hasta que nivel de detalle llegaremos. Podemos no complicarnos la vida y elegir **Recognize** > **Recognize All Pages** (**Ctrl** + **↑** + **R**), dejar que el programa haga lo suyo, no revisar ni corregir nada y seguir adelante. Si así lo disponemos, tal vez podemos saltarnos esta sección completamente y aplicar los métodos del apéndice F. Mi recomendación es que como mínimo se revise la segmentación de las áreas de cada página y que se entrene al menos un poco el patrón. Pienso que la ganancia en precisión vale la pena para el poco tiempo que esto insume.

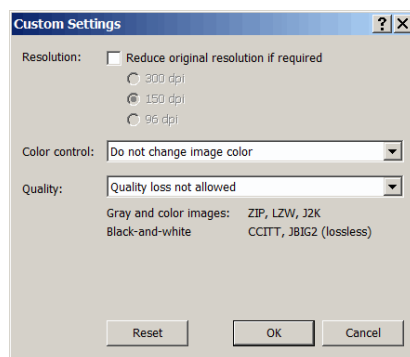
El reconocimiento de páginas se realiza en dos pasos. El primero es de análisis (**Analyze**) y el segundo de reconocimiento propiamente (**Recognize**).

Mediante el análisis, FINEREADER determina las zonas de contenido en la página. Estas pueden ser texto, imagen, tabla, imagen de fondo o área de reconocimiento. En la figura 7.7 se muestra el detalle de la barra con las herramientas para crear y editar las zonas. Además de los tipos de zona para dibujar tiene herramientas de selección, vistazo, eliminación y reordenación de zonas.⁵

⁵ En inglés, las de dibujo son: **Draw Text Area**, **Draw Picture Area**,



(a) Ventana principal.



(b) Custom Settings.

FIGURA 7.4. Opciones de FINEREADER. Pestaña **Format Settings** » **PDF**.

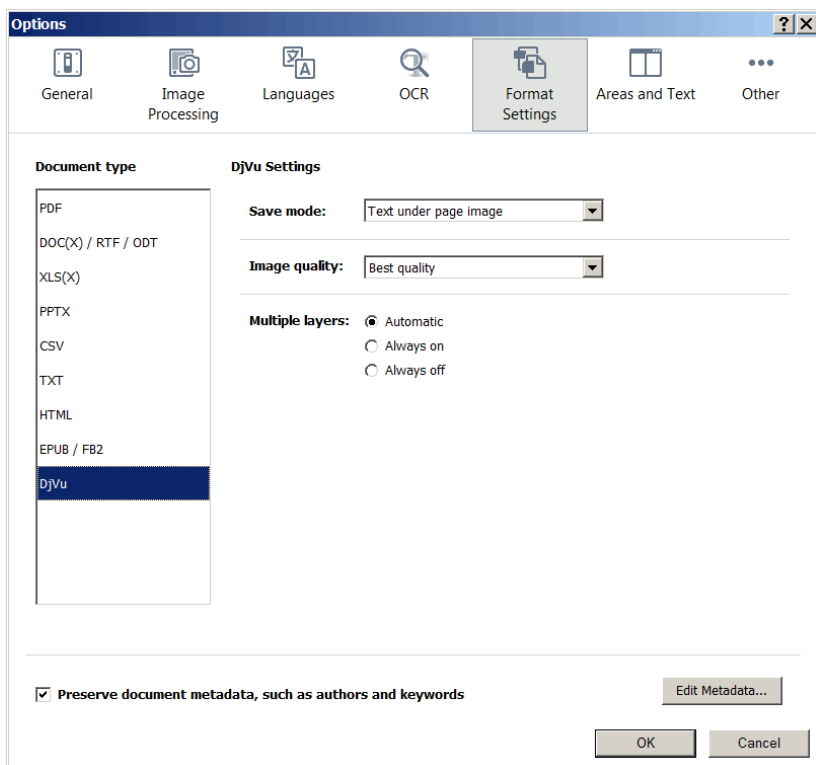


FIGURA 7.5. Opciones de FINEREADER. Pestaña **Format Settings** > **DjVu**.

Salvo que el contenido sea muy homogéneo, por ejemplo: solo texto o texto con alguna imagen, el algoritmo de segmentación de zonas no funciona muy bien y estas deben ser delimitadas manualmente.

Normalmente las creo como **Recognition areas** (áreas de reconocimiento), para que FINEREADER determine el tipo de contenido y ajuste el tamaño. Esto no es estrictamente necesario y el área puede ser más

Draw Table Area, **Draw Background Picture Area** y **Draw Recognition Area**. Y las de edición: **Select Objects**, **Pan Around the Page**, **Delete Area** y **Reorder Areas**. También existe una **Barcode area**, para reconocer códigos de barras. No es muy útil y solo será mencionada aquí.

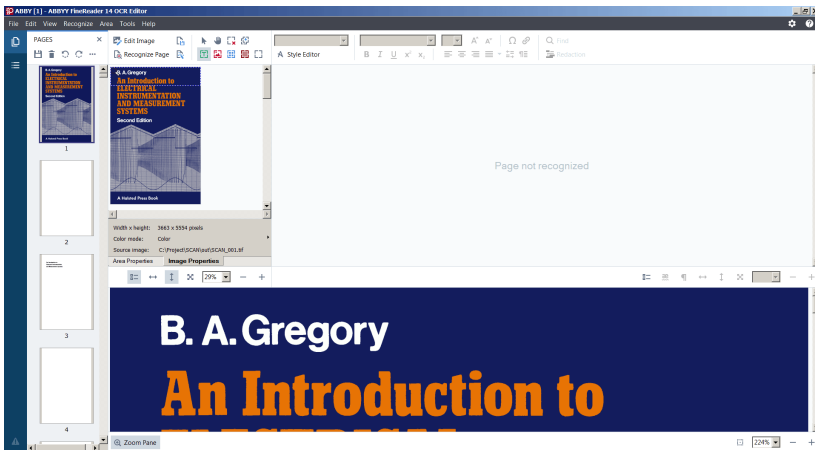


FIGURA 7.6. Ventana principal de FINEREADER una vez cargadas las páginas. Se ha personalizado la disposición de los paneles respecto a los valores por defecto. La columna de la izquierda es el **Navigation Pane**, que muestra las miniaturas de todas las páginas. Hacia la derecha y arriba está el **Image Pane**, que muestra la página actual. A la derecha de este está el **Text Pane**, con el texto reconocido. Debajo de estos dos últimos tenemos el **Zoom Pane**, que muestra un zoom del **Image Pane**.



FIGURA 7.7. Herramientas de edición de zonas de FINEREADER.

grande que el contenido que se busca reconocer, mientras no se solape con ninguna otra. Puede establecerse el tipo (generalmente **Text**) manualmente durante la creación.

Se pueden dibujar tantas áreas como uno quiera. Para nuestro procedimiento usaremos solo las de texto y las de tablas. No será necesario marcar imágenes (aún si no las hubiéramos separado con SCAN TAILOR), pues elegimos la opción **Text under the page image**, que preservará las imágenes originales e insertará el texto reconocido debajo de las mismas.

Colocar todo el texto dentro de una sola área o usar varias áreas para segmentar el mismo contenido es indistinto desde el punto de vista del resultado final. Si vamos a intentar corregir los errores que veamos, puede convenir no tener todo en una sola área. Esto porque a veces editamos algo manualmente, pero mas abajo nos damos cuenta que queremos reconocer el texto del área de nuevo, porque entrenamos el patrón o lo que sea. Pero de hacerlo perdemos los cambios manuales hechos al principio. Si tenemos varias áreas es más difícil que esto suceda.

Una vez que las áreas fueron reconocidas, ya sea presionando **Recognize** **Analyze Page** o manualmente, estamos en condiciones de reconocer el texto. Esto lo hacemos con **Recognize** **Recognize Page** (**Ctrl** + **R**).⁶ Así, el programa reconoce el texto en las áreas marcadas y lo presenta en el panel de texto. Para reconocer solo algunas áreas, las seleccionamos y hacemos **☞** **Recognize** (o usamos el atajo **Ctrl** + **↑** + **B**).

En el panel de texto podemos ver el resultado del reconocimiento y realizar cambios. Conviene que estos no sean mayores, porque en el PDF se puede perder el sincronismo entre la imagen que vemos y el texto reconocido debajo. Ante la duda, hacemos una «vista previa» como se explica más adelante.

⁶ Si para la página actual tenemos confianza en que el algoritmo automático de segmentación hará un buen trabajo podemos dar **Recognize Page** directamente, que hace los dos pasos de una sola vez.

7.3.1. Errores y recomendaciones

A continuación hablaremos sobre algunos de los errores de reconocimiento comunes que vamos a encontrar y daremos algunas recomendaciones. Como siempre, se trata de lineamientos con los que he trabajado y que considero que funcionan bien, pero cada uno deberá experimentar y encontrar el punto de trabajo con el que se siente cómodo.

Pestaña Areas and Text

FINEREADER permite resaltar los errores o dudas de reconocimiento que tenga, de forma de facilitar la corrección manual. Para la configuración de este comportamiento tenemos la pestaña **Areas and Text** del menú de opciones (fig. 7.8), donde podemos elegir qué resaltar y con qué colores. Recomiendo elegir un color fácilmente distinguible para los caracteres dudosos.⁷

Reconocimiento por página o por área

Por alguna razón, muchas veces los resultados de reconocer toda la página de una vez y de hacerlo área por área difieren. Si hay un área muy problemática, podemos intentar reconocerla nuevamente, para ver si al ser individualizada los resultados mejoran.

Nuevos errores y actualización del proyecto

A veces durante el proceso de reconocimiento descubrimos errores que pasamos por alto durante el proceso de limpieza y ajuste. Pueden ser manchas, caracteres mal impresos o cualquier cosa que no queramos que quede en nuestro documento final. Nótese por ejemplo las manchas en la figura 7.9.

Entonces, volvemos a SCAN TAILOR (y/o GIMP), hacemos las modificaciones pertinentes y nos vemos ante el problema de actualizar el proyecto de FINEREADER. Podemos hacer  **Page Properties** en la página correspondiente en el panel de navegación y presionar en

⁷ El cian que viene configurado por defecto puede confundirse con el fondo blanco, por lo que normalmente uso algún rojo.

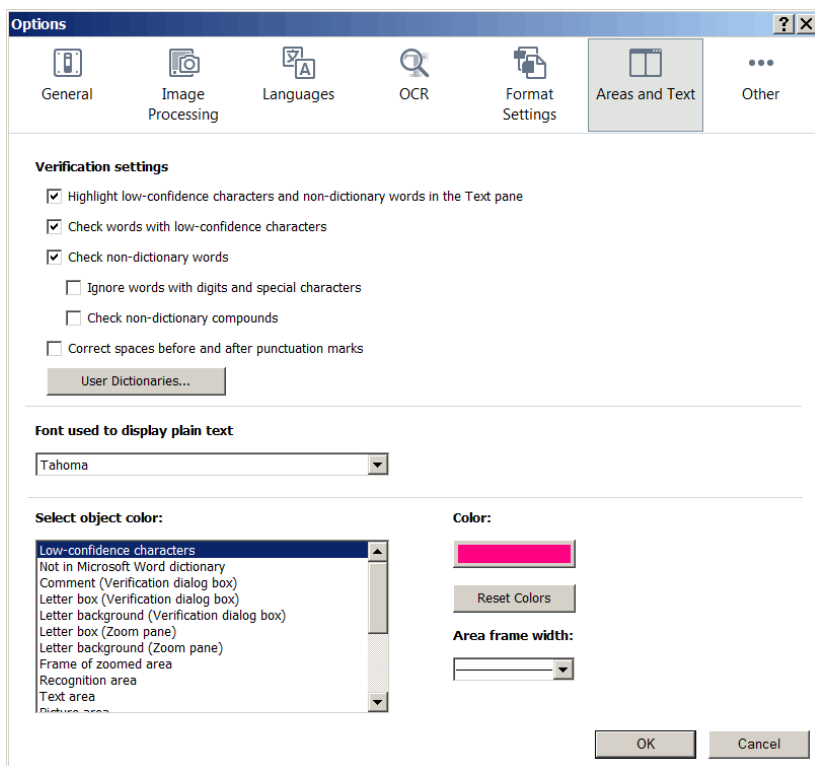


FIGURA 7.8. Opciones de FINEREADER. Pestaña **Areas and Text**.

⁴ Consider figure 1.7 where the fixed segments have been connected to a motor-driven rotary switch. One revolution of the switch contact will sample the

Consider figure 1.7 where the fixed segments have been connected to a motor-driven rotary switch. One revolution of the switch contact will sample the

FIGURA 7.9. Dos manchas no detectadas durante el paso de limpieza y guion dejado fuera del área de reconocimiento. Se muestra la sección del área de texto correspondiente.

Reopen Page, seleccionando la página que acabamos de corregir. Este procedimiento es un poco tedioso, pero para pocas páginas está bien.

Si las ediciones son frecuentes, podemos ahorrar algo de tiempo mientras no cambiemos las dimensiones de la página,⁸ ejecutando el script 7.1.⁹ Este nos consulta por la página que queremos substituir y lo hace rápidamente. Su funcionamiento es un poco delicado, por lo que si no funciona bien recomiendo leer el apéndice G.

replaceInABBY.py

```
# This script replaces a B&W page in an Abby
↳ FineReader 14 project with a new one from the out
↳ folder.

import os, shutil

print('Which page should I replace?')
page = input()

oldFile = '../ABBY/' + page.zfill(4) +
↳ '/Image/bwPage.frdatt'
newFile = '../SCAN/out/SCAN_' + page.zfill(3) +
↳ '.tif'

# Copy (and overwrite) the new file.
shutil.copy(newFile, oldFile)
```

SCRIPT 7.1. Reemplazo de imagen bitonal en FINEREADER 14.

Ajuste del alfabeto

El alfabeto es el conjunto de caracteres que el programa asume que pueden aparecer en un documento dado. Al crear nuestro idioma se elije por defecto un conjunto más o menos razonable de letras, números y

⁸ Esto no debería ocurrir en este punto.

⁹ Para que funcione debemos haber mantenido la numeración correcta de las páginas y no haber usado el método anterior con esta página. Además, la imagen debe ser bitonal y FINEREADER no debe tenerla abierta.

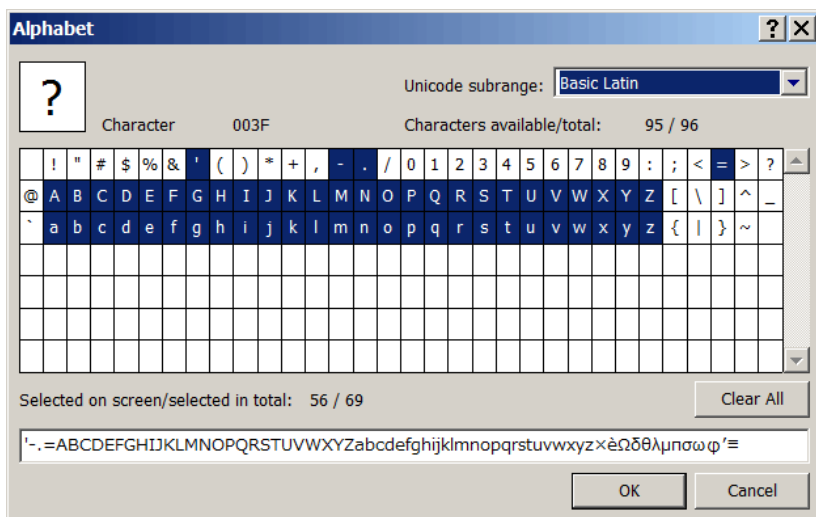


FIGURA 7.10. Edición del alfabeto.

signos de puntuación. FINEREADER solo reconocerá aquellos caracteres que pertenezcan a este alfabeto.

Si nuestro documento utiliza más caracteres,¹⁰ debemos agregarlos al diccionario y preferentemente entrenar el patrón de usuario para facilitar su reconocimiento.

Esto lo haremos editando las propiedades de nuestro idioma (ver fig. 7.3(b)). Allí, presionando , podemos ver y editar los caracteres incluidos.¹¹

El cuadro de diálogo se muestra en la figura 7.10. En el menú desplegable **Unicode subrange** tenemos páginas con los diferentes conjuntos de caracteres para seleccionar.¹²

¹⁰ Por ejemplo, nuestro material contiene algunas letras del alfabeto griego.

¹¹ En realidad básicamente solo las letras y números. **Advanced** despliega un menú similar, pero que además incluye los distintos símbolos y signos de puntuación. En algún caso podríamos necesitar trabajar con él (es análogo).

¹² Normalmente solo uso caracteres de las páginas: **Basic Latin**, **Latin-1 Supplement**, **Basic Greek**, **General Punctuation** y **Mathematical Operators**, si bien hay algunos de otras páginas seleccionados por defecto. Esto dependerá mucho de lo que haya impreso en el material que estemos reconociendo.

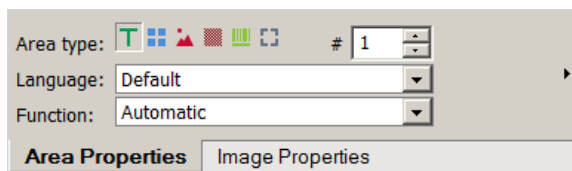


FIGURA 7.11. Pestaña **Area Properties** de FINEREADER.

Podría ocurrir que nuestro alfabeto tuviera caracteres que no aparecen en nuestro material, pero que el software está confundiendo. Por ejemplo, es frecuente la confusión entre «*» y «●». Si este último símbolo no aparece entonces podemos borrarlo del alfabeto (o agregarlo a los caracteres prohibidos en el menú **Advanced**). Esto no nos asegura el reconocimiento que esperamos (todavía podría confundirlo con otro), pero en general mejora los resultados.

La edición del diccionario puede hacerse parte por parte. A veces agregamos un caracter porque se utiliza en una parte del libro, pero más adelante, aunque no aparece más, es erróneamente reconocido. Llegado a este punto se puede quitar del alfabeto (y volver a agregarse de ser necesario más adelante).

Numeración de las zonas

La numeración de las zonas es importante si queremos que la selección de texto sea natural en la página.

El número del area seleccionada lo podemos ver en la pestaña **Area Properties** que se encuentra debajo del **Image Pane** (fig. 7.11). Allí además podemos cambiar el tipo de área, el idioma y haciendo clic en  accedemos a la orientación y otras opciones.

Supongamos que tenemos tres zonas que se encuentran una encima de la otra. Si de arriba hacia abajo las ordenamos 1-3-2, cuando en el documento final queramos seleccionar texto de distintas áreas a la vez, veremos que no ocurre como desearíamos. Recomendando experimentar un poco cambiando los números y viendo el resultado para entender de que estamos hablando.

Cuando el contenido es complejo (texto, figuras con texto que queremos reconocer) puede ser difícil decidir como ordenar las áreas. Sugiero

idear un criterio y aplicarlo consistentemente a todo el documento.

Texto rotado

A veces parte del texto impreso en una página tiene una orientación distinta del resto. Por ejemplo, es común que en las gráficas, la etiqueta del eje vertical se imprima rotada 90°. Si el texto en el área que queremos reconocer se encuentra rotado 90°, 180° o 270°, debemos seleccionar el tipo de área **Text** y desde el menú de propiedades del área indicar que orientación tiene para que el reconocimiento funcione bien (**Rotated left**, **Rotated right** o **Rotated 180 degrees**). Para texto a 45° u otro ángulo no recto no disponemos de opciones, por lo que no podrá ser reconocido.

Mejor diccionario que edición manual

Si hay alguna palabra mal reconocida es probable que esta no esté en el diccionario. Si es así, FINEREADER la subraya y haciendo  sobre ella nos presenta una lista de posibles candidatas tomadas del diccionario. Desde el punto de vista del sincronismo, es mejor corregir la palabra de esta manera que borrando y escribiendo manualmente.

Área no abarca todo el contenido

En algunas áreas, sobre todo aquellas en donde el texto no está justificado, FINEREADER tiende a dejar fuera algunos caracteres, especialmente los guiones o signos de puntuación (ver fig. 7.9). En estos casos debemos modificar el área para que pase a incluir todos los contenidos.

Errores comunes

Para los errores que se repiten comúnmente, por ejemplo cuando FINEREADER confunde «R» por «/?», podemos ir tomando nota en un archivo de texto. Al terminar de reconocer todas las páginas, usamos la función de buscar (**Ctrl** + **F**) para encontrarlos y corregirlos todos de una sola vez.

(ii) *Detemination error* This is the uncertainty in the indicated value due to

(ii) *Detemination error* This is the uncertainty in the indicated value due to

(a) «Detemination» en vez de «Determination».

homogenities in the tape coating. At long wavelengths (low frequencies) the

homogenities in the tape coating. At long wavelengths (low frequencies) the

(b) «homogenities» en vez de «homogeneities».

Figure 3.3 outlines the main feaures of a three-dial potentiometer with a

Figure 3.3 outlines the main feaures of a three-dial potentiometer with a

(c) «feaures» en vez de «features».

FIGURA 7.12. Algunas erratas en el libro de ejemplo.

Erratas en el original

A veces el material contiene erratas que pueden ser detectadas porque FINEREADER resalta con subrayado rojo las palabras que no conoce. En los casos de la figura 7.12 se trata de palabras a las que les faltó una letra. Para corregirlas podemos insertar manualmente la letra faltante o, aún mejor, hacer  sobre la palabra subrayada y elegir la correcta dentro de la lista de sugerencias que se nos ofrece.

Otra alternativa es editar el original con GIMP para corregir la errata.

Vista previa

Podemos ver como va a quedar una página haciendo  Save as Searchable PDF Document en la correspondiente página en el panel de navegación. De esta forma podemos ver como va a quedar el texto en en esa página. Si tenemos alguna duda, conviene abrir el PDF creado y seleccionar el texto de interés para ver que esté alineado con la imagen. También podemos copiar y pegar en el NOTEPAD + +, para ver que haya sido bien reconocido.

7.3.2. Patrones

FINEREADER contiene patrones por defecto que han sido entrenados con muchos tipos de documentos. En general funcionan bien, pero es

conveniente ayudarlo un poco entrenando un patrón personalizado para el documento presente.

El proceso de entrenamiento no es difícil, pero para los detalles refiero al lector al manual de usuario de FINEREADER. A continuación veremos algunos consejos y buenas prácticas respecto al trabajo con patrones.

Configuración

Para que el patrón entrenado se use hay que elegir ☐ Use user patterns con ☒ Also use built-in patterns, de manera de que se puedan complementar (ver figura 7.13). Solo el patrón personalizado es mala idea, pues en mi experiencia no funciona bien.

Qué entrenar

Debe tenerse en cuenta que el patrón debe corresponder a un solo tipo y tamaño de fuente. Es decir, será perjudicial entrenarlo con texto principal y con el de los títulos, tapas, etc. Sí soporta bien variantes de la misma fuente, como pueden ser negrita, cursiva, superíndice o subíndice.

De todas formas, debemos ser cuidadosos. A veces sucede que el programa reconoce bien alguna letra y al entrenar un poco el patrón deja de hacerlo, o comienza a confundirla con otra. En mi experiencia, es conveniente no entrenar la letra «l», pues una vez que lo hacemos comienza a confundir los «1» con estas, y las secciones muchas veces pasan a decir «l.l» en lugar de «1.1», o similares. Lo mismo puede suceder con otros caracteres. Ante la duda se puede guardar el patrón presionando (fig. 7.13) y probar editarlo, eliminando o agregando algún carácter para ver como reacciona el programa. En caso de que no se obtengan los resultados buscados, se puede restaurar presionando en .

Ligaduras

En tipografía existen las así llamadas ligaduras, que ocurren cuando dos o más caracteres que se imprimen juntos quedan unidos. Son liga-

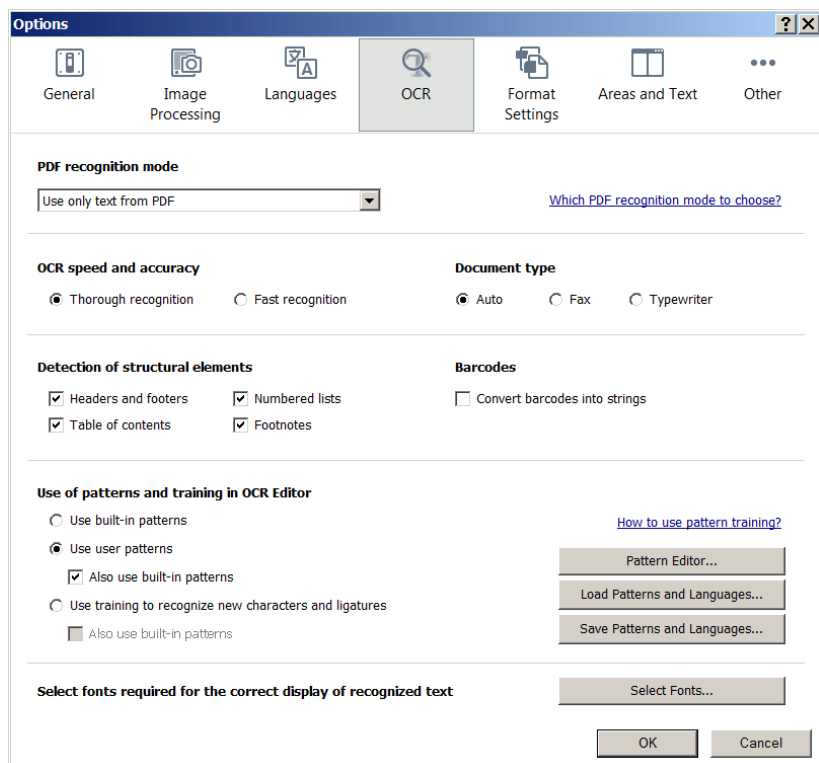


FIGURA 7.13. Opciones de FINEREADER. Pestaña OCR.

duras frecuentes (en el alfabeto latino) «fi», «ff», «ffi» y «ffl», aunque puede haber otras.

Podemos entrenar nuestro patrón para reconocer ciertas ligaduras. Si esta no existe previamente FINEREADER ofrece incluirla en el alfabeto. Véase la figura 7.14.

A veces es útil entrenar como ligaduras pares o ternas de letras que en realidad no lo son. Por ejemplo en los subíndices. A veces FINEREADER tiene dificultad para reconocer el subíndice de A_{bc} (o similar). En dicho caso podría servir entrenar el patrón para que reconozca bc como una ligadura, aunque estrictamente no lo sea.

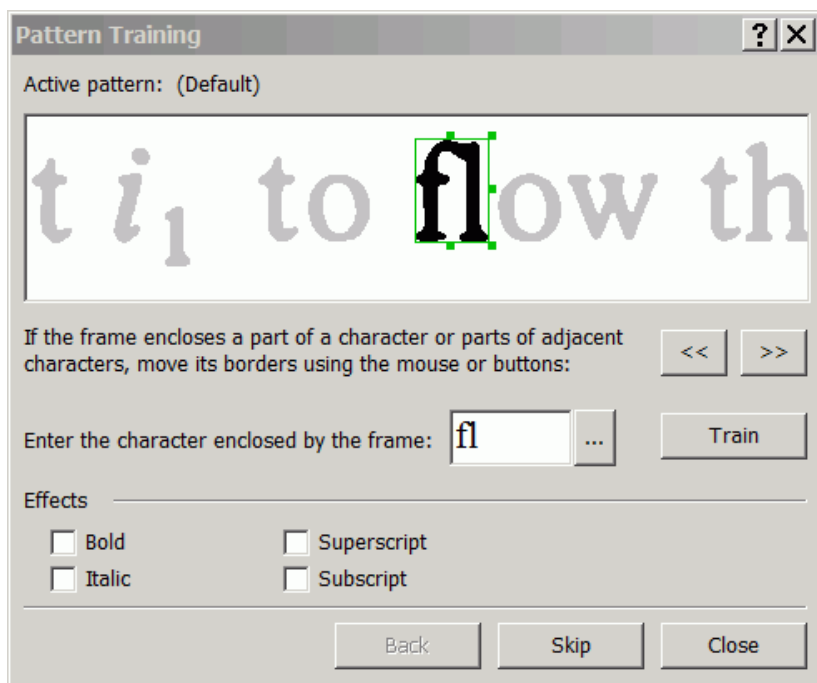


FIGURA 7.14. Entrenamiento y creación de la ligadura «fl» en FINEReader.

Patrón + diccionario

El reconocimiento se basa fuertemente en el diccionario del idioma correspondiente. Si nuestro material contiene palabras que no están en el diccionario, FINEREADER tiene más probabilidad de reconocerlas erróneamente. Para ayudarlo, podemos incluir estas palabras en el diccionario a medida que van apareciendo. Inclusive ayuda agregar siglas o claves del tipo «l.s.d.» (*least significant digit*, dígito menos significativo), «1.6.» (para los números de sección o de ecuación), «LED», etc.

Con un patrón suficientemente entrenado y un diccionario que contenga la mayoría de las palabras que aparecen en nuestro material solo tendremos que corregir algunos errores en cada página.

Caracteres dudosos

Si vemos que resalta muy seguido un carácter como dudoso pero aún así lo reconoce bien, no debemos preocuparnos por entrenar el patrón para que deje de ser dudoso. A veces arreglamos eso pero empeoramos otro aspecto que funcionaba bien.

7.4. Conclusiones

Una vez reconocidas todas las páginas (inclusive las vacías) estamos en condiciones de generar nuestros documentos.

Seleccionamos entonces  **File**  **Save As**  **Searchable PDF Document**, ponemos el nombre  **SCAN.pdf** y lo guardamos en la carpeta  **djv**.¹³ Este será nuestro PDF base, sobre el que agregaremos los marcadores, metadatos y demás.

Análogamente, guardaremos un archivo  **OCR.djvu**. Este no será de base, sino que lo usaremos solo para extraerle la capa de texto reconocida e insertarla en nuestro DJVU final (ver 9.4).

¹³ El programa se tomará un tiempo para completar la exportación.

Creación del pdf

Llegó el momento de crear nuestro documento PDF final. Básicamente, debemos insertar las imágenes (si es que las separamos), agregar los marcadores, corregir las etiquetas de las páginas, incluir los metadatos, eventualmente optimizar el documento y guardar en formato PDF/A. Fácil, ¿no?

8.1. Insertar imágenes

Si decidimos separar las imágenes del resto del contenido como se explica en 6.2.6, debemos ahora insertarlas nuevamente en el documento con el texto ya reconocido.

8.1.1. Conversión con ImageMagick

Tenemos dos scripts para insertar las imágenes. Ambos utilizan `IMAGEMAGICK` para convertir los archivos TIFF sin pérdidas en JPEG comprimidos con pérdidas encapsulados en PDF. Esto lo hacen mediante líneas del tipo `magick convert SCAN_XXX.tif -transparent white -compress jpeg -quality 40 SCAN_XXX.tif.pdf`, donde:

- `magick convert` significa convertir con `IMAGEMAGICK`;
- `SCAN_XXX.tif` es el TIFF original;

- `-transparent white` convierte el blanco en transparente;
- `-compress jpeg` impone compresión JPEG con pérdidas;
- `-quality 40` es el nivel de compresión (va de 1 a 100);¹ y
- `SCAN_XXX.tif.pdf` es el archivo PDF de salida.

Luego superpone las imágenes sobre las páginas correspondientes. Para que esta superposición no opaque el resto del contenido es esencial la opción `-transparent white` de IMAGEMAGICK.² Cada uno difiere en la forma en la que insertan esas páginas en nuestro documento base.

8.1.2. Usando qpdf

El script 8.1 lo hace invocando la aplicación QPDF, mediante líneas del tipo `qpdf --overlay SCAN_XXX.tif.pdf --from=1 --to=XXX -- SCAN_INS.pdf`, donde:

- `qpdf` invoca al programa;
- `--overlay` indica superposición;
- `SCAN_XXX.tif.pdf` es el documento estampa;
- `--from=1` es el número de página que queremos imprimir (solo hay una);
- `--to=XXX` es el número de página donde debe imprimirse; y
- `SCAN_INS.pdf` es el nombre del documento que se va a imprimir.

insertImagesqpdf.py

```
# This script pastes the separated photos in the PDF.
```

```
import os, shutil, glob, re
```

```
backgroundDir = '../SCAN/out/background/'
```

```
SCAN = '../djv/SCAN.pdf'
```

```
SCAN_INS = '../djv/SCAN_INS.pdf'
```

```
TEMP = '../djv/TEMP.pdf'
```

¹ En este caso elegí un valor de 40, pero deberemos buscar el valor que consideremos óptimo en cada caso.

² Recomendando probar qué ocurre si se omite.

insertImagesqpdf.py (cont.)

```

# Create new SCAN_INS.pdf to work with (eventually
↳ overwriting the old one).
shutil.copy(SCAN, SCAN_INS)

namesList = [os.path.basename(x) for x in
↳ glob.glob(backgroundDir + '*.tif')];

for filename in namesList:
    # Call ImageMagick to create the
    ↳ transparent-white pdf of the page.
    os.system('magick convert ' + backgroundDir +
    ↳ filename + ' -transparent white -compress
    ↳ jpeg -quality 40 ' + backgroundDir +
    ↳ filename + '.pdf ')
    print(filename + '.pdf created!')
    # Call qpdf to paste that pdf in the
    ↳ corresponding page of SCAN_INS, but save
    ↳ it as TEMP (can't specify the same file
    ↳ as both the input and the output, thus
    ↳ the need for TEMP).
    os.system('qpdf --overlay ' + backgroundDir +
    ↳ filename + '.pdf --from=1 --to=' +
    ↳ re.search(r'\d+', filename).group(0) + '
    ↳ -- ' + SCAN_INS + ' ' + TEMP)
    # Make TEMP the new SCAN_INS and delete the
    ↳ old TEMP and pdf files.
    shutil.copy(TEMP, SCAN_INS)
    os.remove(TEMP)
    os.remove(backgroundDir + filename + '.pdf')
    print(filename + '.pdf inserted!')

```

SCRIPT 8.1. Inserción de fotos usando QPDF.

Su ejecución puede tomar un par de minutos. Al terminar tendremos

un archivo  SCAN_INS.pdf con las imágenes insertadas.

8.1.3. Usando Acrobat DC

Por otra parte, el script 8.2,³ además de invocar a `IMAGEMAGICK` como ya se explicó, crea a su vez un script de *Adobe Javascript* para insertar las páginas usando `ACROBAT DC` [31, 32]. Este script se guarda en el archivo  insertImagesScript.txt y contiene líneas del tipo `this.addWatermarkFromFile({cDIPath: "/C/Project/SCAN/out/background/SCAN_XXX.tif.pdf", nSourcePage: 0, nStart: XXX-1});`, donde:

- `this.addWatermarkFromFile` es la función que estampa imágenes, esta función usa los siguientes parámetros:
 - `cDIPath` ruta del documento que queremos estampar;
 - `nSourcePage` página, indizada en 0, del documento anterior que queremos estampar; y
 - `nStart` página, indizada en 0, del documento de destino.

insertImagesAcrobat.py

```
# This script generates an new Adobe Javascript
↳ script that will be applied with Acrobat to paste
↳ the separated photos in the PDF.

import os, re, glob

backgroundDir = '../SCAN/out/background/'
insertImages = 'insertImages.js'
SCAN_INS = '../djv/SCAN_INS.pdf'

# Delete old file (if it exists).
if os.path.isfile(insertImages):
    os.remove(insertImages)
```

³ Lamentablemente no he conseguido que el script 8.2 funcione con rutas de directorios relativas, por lo que el lector deberá cambiar `/C/Project/SCAN/out/background/` por el directorio correspondiente, si es que difiere.

insertImagesAcrobat.py (cont.)

```

namesList = [os.path.basename(x) for x in
↳ glob.glob(backgroundDir + '*.tif')]

with open(insertImages, 'w') as outputFile:
    for filename in namesList:
        # Call ImageMagick to create the
        ↳ transparent-white pdf of the
        ↳ page.
        os.system('magick convert ' +
        ↳ backgroundDir + filename + '
        ↳ -transparent white -compress jpeg
        ↳ -quality 40 ' + backgroundDir +
        ↳ filename + '.pdf ')
        print(filename + '.pdf created!')
        # Save the corresponding line to the
        ↳ script.
        print('this.addWatermarkFromFile({cDIPath:
        ↳ "/C/Project/SCAN/out/background/'
        ↳ + filename + '.pdf", nSourcePage:
        ↳ 0, nStart:',
        ↳ str(int(re.search(r'\d+',
        ↳ filename).group(0))-1) + '});',
        ↳ file=outputFile)
    print('this.saveAs("' + SCAN_INS + '");',
    ↳ file=outputFile)

```

SCRIPT 8.2. Inserción de fotos usando ACROBAT DC.

Para aplicarlo, debemos abrir el documento base  SCAN.pdf con ACROBAT DC e iniciar la consola de javascript. Se puede hacer eligiendo    o presionando  + . Deberá aparecer una ventana como la de la figura 8.1.

En la consola pegamos el contenido de  insertImagesScript.txt, lo seleccionamos todo y presionamos  + . Luego de algunos

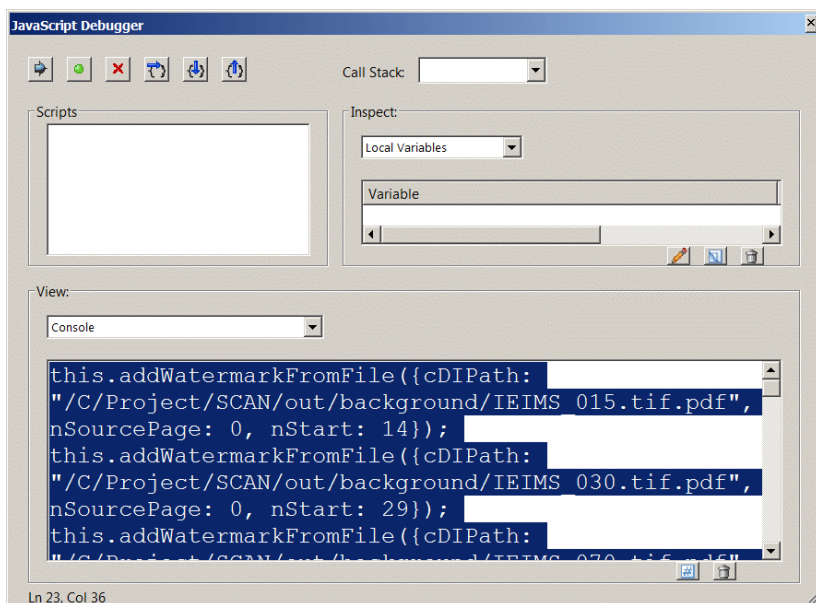


FIGURA 8.1. Consola de Adobe Javascript lista para insertar imágenes.

segundos aparecerá nuestro archivo `SCAN_INS.pdf` con las imágenes insertadas. Si *no* ocurrió ningún error, la consola mostrará el mensaje `undefined`.

8.1.4. Nota: sobre el tamaño final

Descubrí que si bien este método funciona bien, el blanco (convertido en transparente) de los JPEG en algunos casos puede llegar a duplicar el tamaño de las imágenes. De momento lo presento así, pero hay que idear un método más eficiente, que recorte las imágenes (fácil) y las inserte en la posición adecuada en el documento final (difícil). (Acepto sugerencias.)

Título del libro→1
 →Copyright→2
 →Prefacio→3

Capítulo 1→4
 →Sección 1.1→5
 →Sección 1.2→6
 →→Subsección 1.2.1→7

Capítulo 2→8

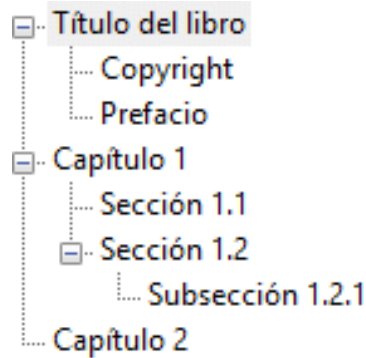


FIGURA 8.2. Formato de marcadores y resultado en el documento.

8.2. Insertar marcadores

Para insertar marcadores comenzaremos abriendo el NOTEPAD++, creando un archivo de texto `insertBookmarks.txt` y pegando allí la tabla de contenidos copiada del PDF.

8.2.1. Formato

Debemos darle formato adecuado a los marcadores para poder insertarlos. El nivel del marcador depende de la cantidad de tabuladores (`(\t)`) que anteceden a su título. Para los marcadores en el nivel superior (0) no habrá tabuladores, para los del nivel 1 habrá 1 tabulador, y así sucesivamente. Los marcadores se anidarán según su propio nivel y el de los demás.

El formato general es: (`→`) Nombre del marcador `→` N.º de página. Podemos ver un ejemplo de marcadores formateados y su correspondiente resultado en la figura 8.2.⁴

8.2.2. Secciones extra

A veces el material está dividido en más subsecciones de las que aparecen en la tabla de contenidos. Si queremos incluirlas deberemos

⁴ Nótese que las líneas en blanco son ignoradas.

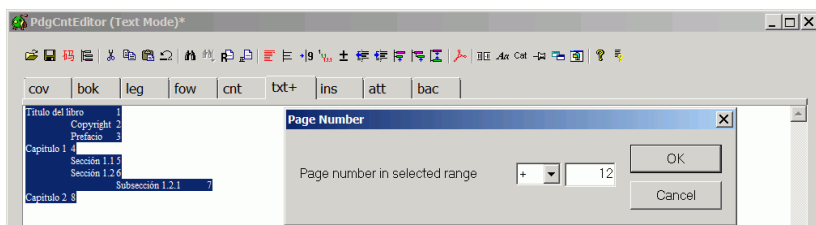


FIGURA 8.3. Ventana del PDGCNTEDITOR. Se presionó .

hacerlo manualmente, recorriendo las páginas, copiando y pegando el texto y agregando el número de página que corresponda.

8.2.3. Corrección de N.ºs de página

Las entradas de la tabla de contenidos están referidas a los números de página impresos en el material, pero nosotros deberemos referirlos a los números de página del documento. Recordemos que en este caso, la página N° 1 es la 13 del documento. Esto significa que debemos hacer un corrimiento de +12 páginas. Para ello podemos usar la aplicación PDGCNTEDITOR.

Pegando los marcadores ya formateados, debemos seleccionar los que queramos corregir y presionar  (Adjust Page Number). En el cuadro de diálogo que aparece (fig. 8.3) ingresamos el desplazamiento que deseamos (en este caso son +12 páginas) y presionamos **OK**. Esos marcadores ajustados los volvemos a pegar en `insertBookmarks.txt`, para tenerlos respaldados.

A su vez, si deseamos agregar marcadores que apuntan a páginas con numerales romanos o letras en vez números (como sucede muchas veces con los prólogos, prefacios, tabla de contenidos, copyright, etc), los referiremos a los números de página del documento.

Una vez terminado el trabajo con los marcadores, el archivo de texto `insertBookmarks.txt` debe tener tantas líneas con el formato de la figura 8.2 como entradas hayamos creado (más algunas líneas vacías tal vez), con los números apuntando a las páginas del documento.

Para insertarlos en el PDF creamos antes una copia de `SCAN_INS.pdf` a la que llamamos `SCAN_INS_BOOK.pdf`. Abrimos el PDGCNTEDITOR, arrastramos y soltamos nuestro PDF sobre el programa y luego copia-

mos los marcadores de `insertBookmarks.txt` y los pegamos allí. Para guardarlos presionamos `Save`.

8.3. Modificar etiquetas de páginas

Hasta ahora, los números de las páginas de nuestro documento no se corresponden con los números impresos. Podemos corregir esto usando ACROBAT DC. En el caso que nos ocupa, las primeras páginas son así:

- | | |
|---|--|
| 1. cubierta del libro; | 9. tabla de contenidos, con «vii» impreso; |
| 2. [página en blanco]; | 10. [página en blanco]; |
| 3. portadilla (o anteportada), con el título de la obra; | 11. prefacio; |
| 4. [página en blanco]; | 12. prefacio, con «x» impreso; |
| 5. portada (título de la obra, subtítulo, edición, autor, editorial, etc.); | 13. introducción, con «1» impreso; |
| | ... |
| 6. página de copyright; | 458. índice, con «446» impreso; |
| 7. tabla de contenidos; | 459. [página en blanco]; |
| 8. tabla de contenidos, con «vi» impreso; | 460. contracubierta; |
| | 461. lomo. |

El primer numeral romano que aparece, «vi», es el de la página 8. Por lo que la cuenta comienza en la página 2 («i») y va hasta la página 12, donde se imprime «x». A partir de la página de introducción comienzan a contarse desde 1 en adelante, hasta la página 458, que tiene impreso el «446». Las restantes tres páginas no tienen número.

Vamos a crear entonces las siguientes etiquetas:⁵

- | | |
|--------------|----------------------|
| 1. Cubierta; | ... |
| 2. FFEP; | 458. 446; |
| 3. i; | 459. RFEP; |
| ... | 460. Contracubierta; |
| 12. x; | 461. Lomo. |
| 13. 1; | |

⁵ Para algunas páginas vacías usaré las etiquetas FFEP y RFEP (del inglés: *Front Free End Paper* y *Rear Free End Paper*). Pueden elegirse otras.

Abrimos  SCAN_INS_BOOK.pdf en ACROBAT DC y en el panel izquierdo presionamos en **Page Thumbnails**, donde podremos ver todas las miniaturas. Allí podemos seleccionar páginas o rangos de páginas y haciendo  **Page Labels** podemos cambiar las etiquetas.

En la figura 8.4 vemos el correspondiente cuadro de diálogo. Para las primeras dos páginas y las tres últimas, elegiremos **None** del menú desplegable **Style**, pues usaremos etiquetas personalizadas, que ingresaremos en **Prefix**. Para las páginas 3-12 elegiremos **Style: i, ii, iii, ...** sin **Prefix**, pues solo queremos números romanos. El resto de las páginas deberían quedar con la numeración 1-446 automáticamente, pero recomiendo revisar y corregir de ser necesario.

8.4. Metadatos y vista inicial

Antes de guardar el documento, sugiero agregar algunos metadatos y configurar la vista inicial. Esto se hace desde el menú **File >> Properties** (**Ctrl** + **D**), pestaña **Description**. Allí sugiero ingresar al menos el nombre de la obra y el autor, y tal vez un comentario que identifique a la persona o grupo que escaneó el material.

La vista inicial la configuramos desde la pestaña **Initial View**. Normalmente uso:

- **Navigation tab**: **Bookmarks and panel page**;
- **Page layout**: **Two-Up (cover page)**; y
- **Magnification**: **Fit Page**.

Otra cosa que podemos modificar es la apariencia de los marcadores. Para ello seleccionamos primero la pestaña **Bookmarks** del panel izquierdo.

Por defecto, todo el árbol se encuentra desplegado. Si lo deseamos, podemos cerrarlo hasta mostrar solo el primer nivel, o lo que prefiramos. Además, haciendo  **Properties** podemos cambiar el tipo de fuente (**Plain**, **Bold**, **Italic**, **Bold&Italic**) y el color. Generalmente uso **Bold** para los elementos del primer nivel y tal vez un color para el título del material.

Finalmente vamos a **File >> Save As...** y guardamos el archivo con el nombre  SCAN_INS_BOOK_PAG.pdf.

Page Numbering [X]

Pages

☐ All

☒ Selected

☐ From: To: of 461

Numbering

☒ Begin new section

Style: ▼

Prefix:

Start:

☐ Extend numbering used in preceding section to selected pages

Sample: 1, 2, 3, ...

OK Cancel

FIGURA 8.4. Ventana de numeración de páginas.

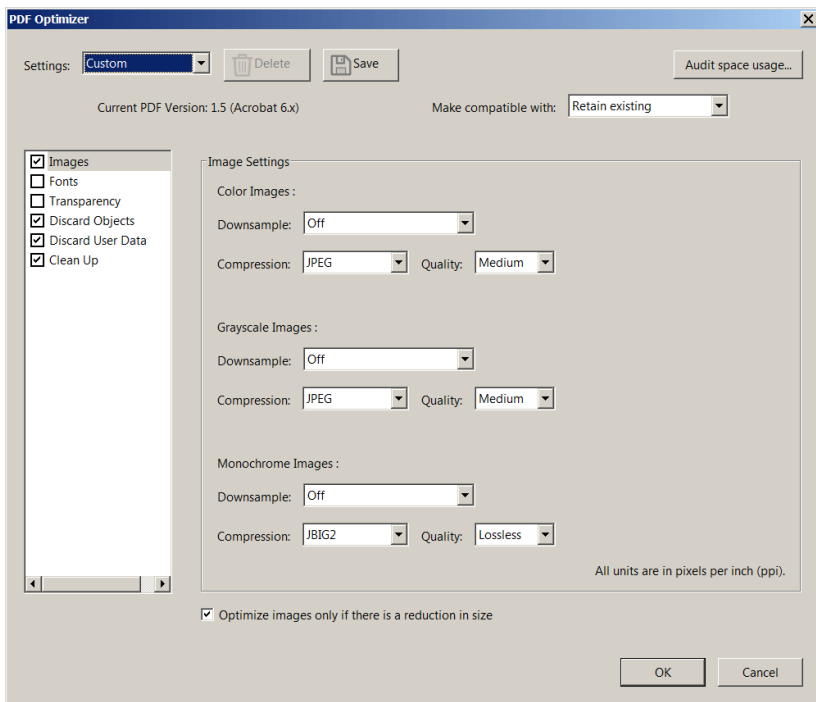


FIGURA 8.5. **PDF Optimizer** de ACROBAT DC. Menú **Images**.

8.5. Optimización

A veces podemos optimizar el PDF para reducir un poco el tamaño sin afectar demasiado la calidad. Para ello seleccionamos **File** > **Save as Other** > **Optimized PDF...** y elegimos las opciones que queramos (ver fig. 8.5).

Las más importantes son las del menú **Images**. Hay que elegir **Downsample** > **Off** para todos los tipos de imágenes, porque de lo contrario estaremos reduciendo las dimensiones, que tanto trabajo pasamos para preservar.

En ☒ **Discard Objects** debemos desmarcar ☐ **Discard document tags** y ☐ **Discard bookmarks**, para no perder el trabajo de la parte anterior.

Lo mismo con ☐ Discard document information and metadata en el menú ☒ Discard User Data.

En todo lo demás hay lugar para la experimentación. Mis opciones son: ☐ Fonts y ☐ Transparency las desmarco siempre. En ☒ Clean Up desmarco ☐ Discard invalid bookmarks, ☐ Discard invalid links y también ☐ Optimize the PDF for fast web view. El resto lo dejo todo marcado.

Una vez elegidas las opciones presionamos y guardamos el documento con el nombre .

8.6. Guardar en pdf/a

Como paso final, podemos guardar el documento como PDF/A. Un PDF/A es un documento PDF común pero que además cumple con el estándar ISO 19005 para el archivado a largo plazo de documentos electrónicos.⁶

Tenemos dos formas de hacerlo. La primera es mediante el menú » » . Allí aparece un cuadro de diálogo para guardar, y con podemos cambiar las opciones.

Yo prefiero usar el menú » y la pestaña (ver fig. 8.6). Allí elegimos , presionamos , elegimos el estándar (recomiendo o) , presionamos nuevamente y por último cambiamos las opciones. En principio marcamos solo y vemos que sucede.

Presionando nos aparece el cuadro de diálogo para guardar. Le ponemos de nombre y esperamos a que termine.

8.7. Verificación

Al guardar como PDF/A ACROBAT DC hace una verificación de cumplimiento del estándar, pero aunque aparezca el mensaje Preflight profile “Convert to PDF” did not find any errors or warnings aún podría tener errores. Como los distintos actores que participan de

⁶ Más información en el correspondiente artículo de Wikipedia [33].

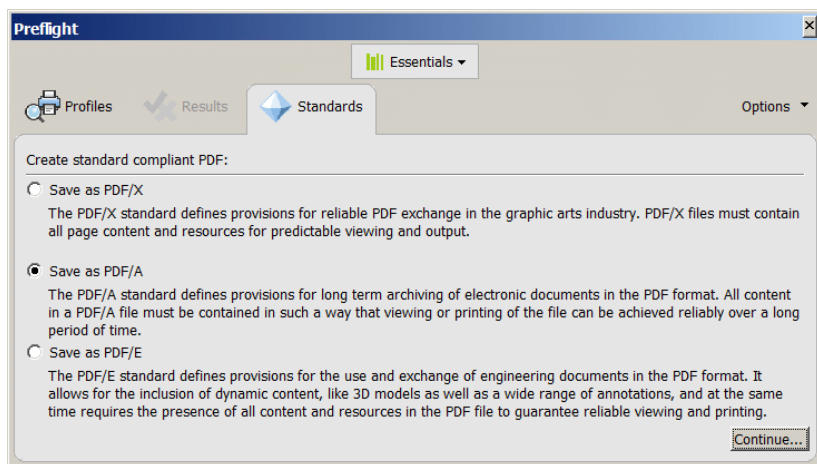


FIGURA 8.6. Menú **Preflight** de ACROBAT DC, pestaña **Standards**.

la industria de creación y verificación de archivos PDF/A tenían discrepancias sobre si tal o cual documento cumplía o no con el estándar, se formó una coalición para desarrollar una herramienta de código libre que fuera la piedra de toque para estos asuntos: se le llamó VERAPDF.

Su utilización es muy simple: iniciamos el programa, hacemos clic en **Choose PDF**, elegimos el archivo que creamos en la sección anterior y presionamos **Execute**. Al terminar sabremos si nuestro archivo efectivamente cumple y si no lo hace qué problemas tiene.⁷

⁷ La solución de estos problemas está fuera del alcance de esta guía.

Creación del djvu

Para crear nuestro documento DJVU podemos usar DJVULIBRE, DOCUMENT EXPRESS, DJVU SMALL o DJVU SMALL MOD. Estos dos últimos ofrecen las aplicaciones principales de DOCUMENT EXPRESS en una interfaz más conveniente, por lo que los preferiremos.

9.1. Configuración de perfiles

DOCUMENT EXPRESS y sus derivados permiten la creación y utilización de perfiles de compresión de DJVU. Perfil es el nombre que se le da al paquete de opciones que se le pasan al codificador para que transforme los TIFF de entrada en (uno o varios) archivo(s) DJVU. Las opciones de cada perfil están (supuestamente) optimizadas para cada tipo de documento y llevan un nombre que intenta ser representativo de ello.

Estos se guardan en la carpeta  profiles del directorio del programa, en un archivo llamado  documenttodjvu.conf. Este es un documento de texto con una sintaxis particular que permite definirlos. Para entenderla basta mirar un poco los perfiles por defecto y los comentarios que dejó el desarrollador. Hay también más información en la documentación de DOCUMENT EXPRESS y de DJVU SMALL MOD.

Para utilizar con mis documentos, yo creo tres perfiles: MyBitonal,

```
#@displayName:MyBitonal (600 dpi)
MyBitonal600: bitonal
dpi=-600
quasilossless=true
pages-per-dict=10
```

(a) Perfil `MyBitonal.`

```
#@displayName:MyPhoto (600 dpi)
MyPhoto600: photo
dpi=-600
```

(b) Perfil `MyPhoto.`

```
#@displayName:MyScan (600 dpi)
MyScan600: scan600
bg-subsample=1
target-subsample=1
pages-per-dict=10
```

(c) Perfil `MyScan.`FIGURA 9.1. **Perfiles personalizados para crear archivos DJVU.**

`MyScan` y `MyPhoto`. Básicamente, `MyBitonal` es para páginas bitonales, `MyPhoto` para fotos y `MyScan` para páginas mixtas, como podrían ser las cubiertas. Fueron creados en base a los perfiles existentes, a la documentación de DOCUMENT EXPRESS y a la información del *Scan&Share Tutorial* [3].

Los nuevos perfiles quedan disponibles usando NOTEPAD++ para añadir las líneas de la figura 9.1 al final de `documenttodjvu.conf`.

El parámetro `pages-per-dict=10` (valor por defecto) se eligió en virtud de lo explicado en [34]. Básicamente, se obtiene un archivo algo más grande (que eligiendo, por ejemplo, `pages-per-dict=1000`) pero

este es más robusto frente a errores.¹

Por otra parte, podemos jugar con el nivel de compresión eligiendo entre las siguientes opciones:

- `aggressive=true`
- `lossy=true`
- `conservative=true`
- `normal=true`
- `quasilossless=true`
- `lossless=true`

He hecho algunas pruebas y, salvo `lossless`, que es totalmente sin pérdidas y da un archivo algo más grande que las otras, las demás alternativas difieren poco en el tamaño final del archivo. Por esta razón me incliné por usar `quasilossless` en el perfil, que según el nombre sería la más fiel al contenido original. He notado que en `aggressive`, por ejemplo, a veces se sustituyen una *o* cursiva por una redonda, o viceversa, y lo mismo puede ocurrir con otros caracteres o símbolos parecidos. Sin embargo, la evaluación de la calidad es bastante subjetiva, por lo que hay lugar para más de una opinión.

Podemos experimentar con las demás opciones para personalizar nuestros propios perfiles si así lo deseamos. El autor de DJVU SMALL MOD creó varios perfiles optimizados para distintos casos. Si vamos a hacer pruebas, recomiendo leer primero la documentación y usarlos como referencia.

9.2. Codificación con DjVu Small

DJVU SMALL ofrece un buen compromiso entre la variedad y flexibilidad de opciones de DOCUMENT EXPRESS y la sencillez de su interfaz gráfica, así como el pequeño tamaño de distribución.²

¹ La documentación de DOCUMENT EXPRESS explica que en la mayoría de los casos la reducción de tamaño comienza a ser marginal a partir de `pages-per-dict=20`, así que podría también usarse ese valor.

² ~200 MB de DOCUMENT EXPRESS contra ~1,5 MB de DJVU SMALL.

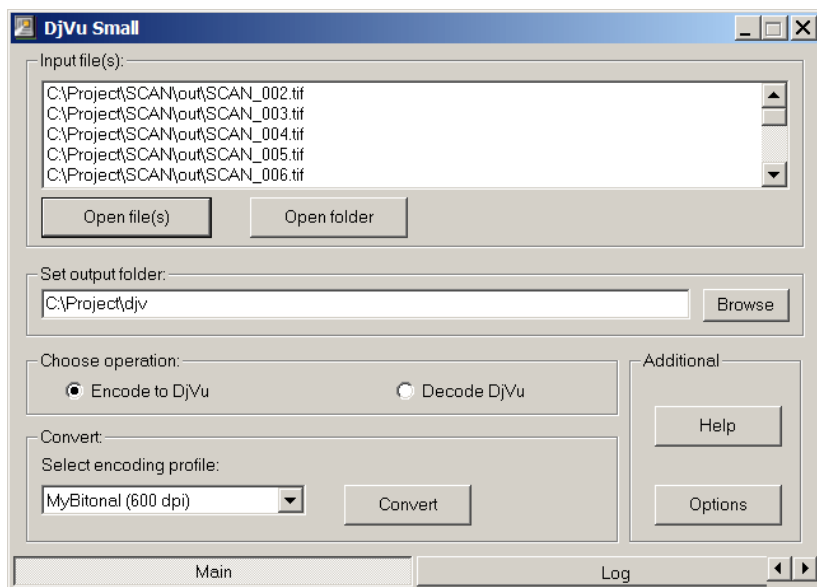


FIGURA 9.2. Ventana principal de DJVU SMALL.

9.2.1. Codificación del cuerpo del documento

Si seguimos la guía correctamente, todas las páginas de nuestro documento salvo las cubiertas deberían estar en blanco y negro. A este conjunto le llamamos cuerpo del texto. En este caso van de la 002 a la 459.

Para comprimirlo abrimos el DJVU SMALL (ver figura 9.2) y usamos **Open File(s)** para agregar las imágenes del cuerpo. Seleccionamos luego el perfil **MyBitonal** del menú desplegable **Select encoding profile**. Elegimos el directorio de salida y presionamos **Convert**. Al archivo resultante le llamaremos **BODY.djvu**.

9.2.2. Codificación de las tapas

Para estas procedemos de la misma manera, solo que utilizamos otro perfil. Habrá que probar un poco para ver cual da el mejor resultado.

Recomiendo intentar con `MyScan`, luego con `Clean (600 dpi)` y tal vez `MyPhoto` u otros. Llamaremos al archivo `COVERS.djvu`.

9.2.3. Unión de cuerpo y tapas

Para unir el cuerpo y las tapas del material ejecutamos el script 9.1, que invoca al programa `djvm.exe` de DJVULIBRE. Este lo que hace es insertar `BODY.djvu` en `COVERS.djvu` a partir de la página `startPage`, que en este caso es la `2`.

bundleDjVu.py

```
# This script inserts the body of the djvu file
↪ between the covers, starting at page startPage.

import os, shutil

SCAN = '../djv/SCAN.djvu'
BODY = '../djv/BODY.djvu'
COVERS = '../djv/COVERS.djvu'

# First page after the front cover.
startPage = '2'

# Delete old file (if it exists).
if os.path.isfile(SCAN):
    os.remove(SCAN)

shutil.copy(COVERS, SCAN)

# Call djvm to insert BODY in between the covers.
os.system('djvm -i ' + SCAN + ' ' + BODY + ' ' +
↪ startPage)
```

SCRIPT 9.1. Unión del cuerpo con las tapas.

9.2.4. Insertar fotos: DjVu Imager

Si utilizamos el modo **Mixed** de SCAN TAILOR, nos está faltando codificar y e insertar las fotos que separamos en nuestro documento DJVU. Para hacerlo utilizaremos el software DJVU IMAGER. Según ha explicado su autor, monday2000, no es más que una GUI para algunas herramientas de DJVULIBRE, pero no por ello deja de ser muy conveniente.

Este usa una versión modificada de la aplicación **f_i_c44.exe** de DJVULIBRE³ para codificar las imágenes, y luego las inserta en nuestro documento.

En la figura 9.3, vemos la ventana principal de DJVU IMAGER. Está dividida principalmente en dos columnas. A la izquierda tiene la lista de archivos que vamos a codificar. A la derecha tenemos, de arriba hacia abajo:

- botones para abrir archivos o directorios y para agregar o quitar elementos de la lista;
- parámetros de calidad de codificación;
- selección de si las opciones actuales aplican para todos los archivos (**All files**) o solo para el actualmente seleccionado (**Current file**), así como un botón para reiniciar las opciones particulares de todos los archivos (**C**);
- botón **Convert**, que convierte a DJVU todos los archivos de la lista, botón **View** (se activa después de dar **Convert**) que abre un DJVU con todos los archivos convertidos y botón **Current** que codifica el archivo actual y lo visualiza;
- botones **Insert in DjVu** (ídem **View**), para insertar las imágenes, **Options** para configurar algunas opciones y **Help** para abrir el archivo de ayuda; y
- rutas de los archivos de origen (**Source**) y destino (**Dest**).

Comenzamos abriendo el programa y configurando las opciones tal cual se muestran en la figura 9.4.

Luego, presionando **Open file(s)**, agregamos a la lista todos los archivos de la carpeta  background.

³ **f_i_c44.exe** utiliza la biblioteca libre **FREEIMAGE** para cargar archivos TIFF directamente. De lo contrario habría que convertir primero a PPM.

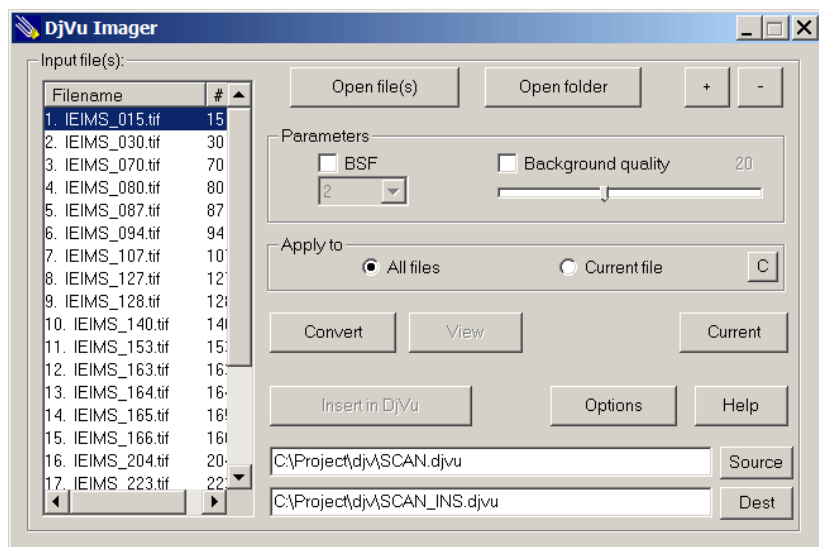


FIGURA 9.3. Ventana principal de DJVU IMAGER lista para convertir.

Debemos elegir ahora las opciones de compresión de las imágenes. **BSF** significa *Background subsample factor* (factor de submuestreo del fondo) y es el factor por el cual se reduce la imagen antes de codificarla. Si por ejemplo nuestra imagen TIFF es 5554 pix por 3663 pix y elegimos ☒ **BSF** , se la escalará a 2777 pix por 1832 pix antes de codificarla. Contrario a la documentación del DJVU IMAGER, yo recomiendo no usar esta opción,⁴ lo que equivale a **BSF** .

La otra, **Background quality** (calidad del fondo), determina la calidad del algoritmo de compresión, aunque no directamente. monday2000 configura algunas opciones del programa `c44.exe` en valores razonables y nos ofrece esa barra deslizante para ajustar el único parámetro que nos deja libre.⁵ Recomendando jugar con esta opción para conseguir el balance de tamaño y calidad deseado. De no activar

⁴ Pasamos bastante trabajo escaneando estas imágenes en 600 dpi como para ahora tirar ese trabajo a la basura.

⁵ Para entender mejor los detalles debemos estar familiarizados con la estructura del formato DJVU, el funcionamiento de `c44.exe` y la documentación de DJVU IMAGER.

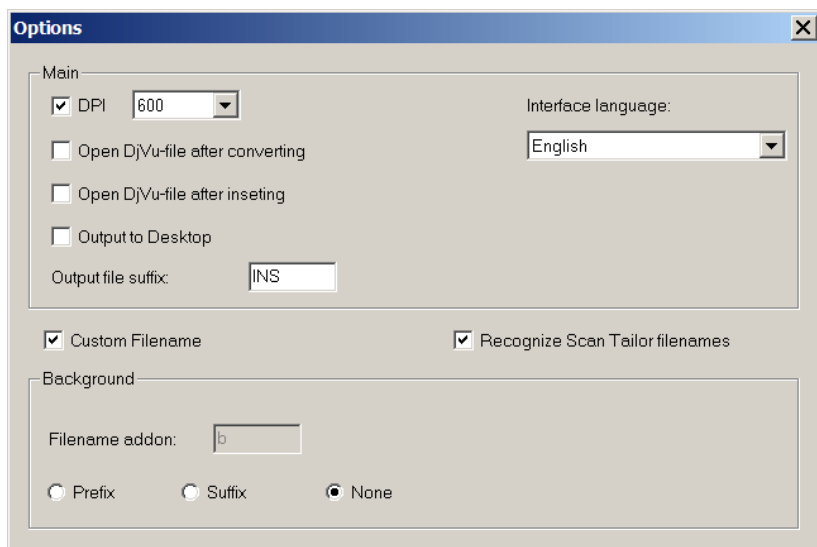


FIGURA 9.4. Ventana de opciones de DJVU IMAGER.

☐ Background quality se utiliza por defecto un valor de aproximadamente ☒ Background quality >> 30.⁶

Podemos ir apreciando el resultado de nuestras opciones en el archivo actual presionando **Current**. Una vez hallamos decidido los parámetros para todos los archivos (o para cada archivo en particular si así lo deseamos) presionamos **Convert** para convertirlos todos. Una vez que termina hacemos **Insert in DjVu** y obtenemos nuestro archivo con las imágenes insertadas.

Alternativa usando DjVu Small

Si no nos conviene el resultado de codificación que da DJVU IMAGER, podemos hacer un poco de trampa y codificar las fotos con DJVU SMALL, usando por ejemplo el perfil **MyPhoto** u otros. El inconveniente es que DJVU IMAGER requiere que cada página a insertar sea un archivo DJVU

⁶ Esto no lo sé con certeza, pero fue el valor que estimé comparando los tamaños de archivo que resultaban activando y no activando la opción.

independiente y DJVU SMALL no dispone de ninguna opción que permita codificar de una sola vez cada archivo por separado, sino que genera un solo DJVU multipágina con todas las imágenes.⁷ Este inconveniente es fácil de resolver: simplemente ejecutamos el script 9.2, que invoca al programa `Ⓜ djvmcvt.exe` de DJVULIBRE para separar las páginas del archivo resultante y reemplaza las generadas por DJVU IMAGER.

separatePhotosDjVu.py

```
# This script uses djvmcvt to unbundle the PHOTOS,
↳ generating one djvu file per page.

import os

PHOTOS = '../djv/PHOTOS.djvu'
# tmp folder of DjVu Imager
tmp = 'C:/SOFT/DjVu Imager/tmp/images/'
index = tmp + 'index.djvu'

# Create folder if it doesn't exist.
os.makedirs(tmp, exist_ok=True)

# Call djvmcvt to create indirect djvu with index
↳ file index.djvu.
os.system('djvmcvt -i ' + PHOTOS + ' "' + tmp + '" "'
↳ + index + '"')
```

SCRIPT 9.2. Separación de las fotos creadas con DJVU SMALL.

Supongamos que usamos DJVU SMALL para generar un archivo `Ⓜ PHOTOS.djvu` con todas las fotos, que guardamos en la carpeta `Ⓜ djv` del proyecto. Entonces, luego de presionar `Convert` ejecutamos el script 9.2⁸ y finalmente presionamos `Insert in DjVu`. De esta manera, engañamos al DJVU IMAGER, haciéndole creer que está insertando los ar-

⁷ La alternativa sería hacerlo uno por uno, lo que podría llevar mucho tiempo.

⁸ El script 9.2 asume que DJVU IMAGER está instalado en `Ⓜ C:⋄SOFT⋄DjVu Imager`. De lo contrario se debe modificar la variable `tmp` para que refleje el directorio correcto.

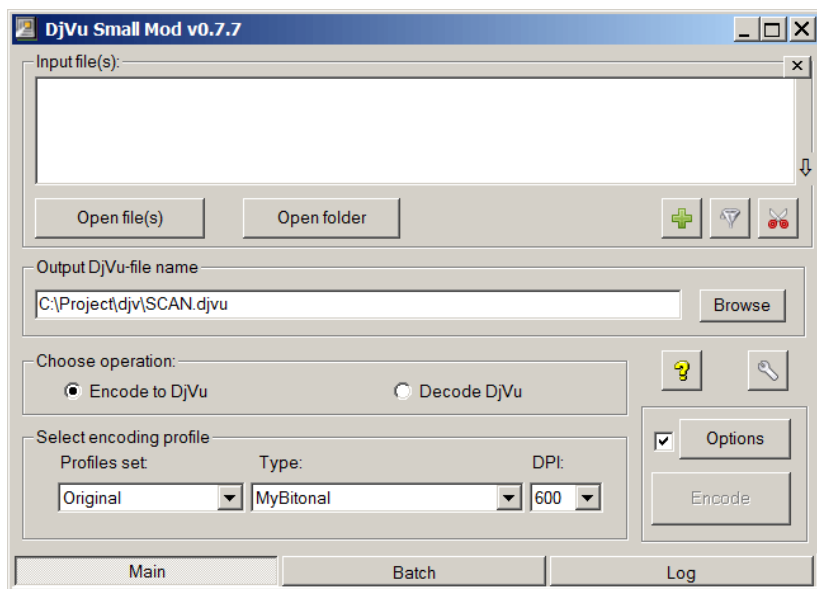


FIGURA 9.5. Ventana principal de DJVU SMALL MOD.

chivos que convirtió, cuando en realidad está insertando los codificados con DJVU SMALL.

9.3. Codificación con DjVu Small Mod

El flujo de trabajo con DJVU SMALL MOD es muy similar al de DJVU SMALL, solo que ahora más funciones se encuentran incluidas en el mismo programa.

En la figura 9.5 podemos ver la ventana principal del programa. De arriba a abajo tenemos:

- la lista de archivos/páginas a procesar;
- botones para agregar o modificar dicha lista;
- la ruta del archivo de salida;
- el tipo de operación (codificar o decodificar) y los botones de ayuda y configuración del programa;

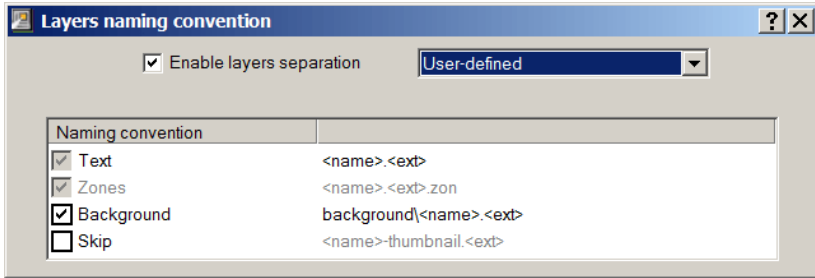


FIGURA 9.6. Ventana de configuración de segmentación manual de DJVU SMALL MOD.

- el perfil a usar y los botones para elegir las opciones de codificación/decodificación y comenzar; y
- botones para cambiar entre la ventana principal, la de procesamiento por lotes y el log.

Los perfiles están agrupados en categorías que se pueden seleccionar desde el menú desplegable `Profiles Set`. Las originales de DOCUMENT EXPRESS están agrupadas en `Original` y el autor de DJVU SMALL MOD creó varios perfiles personalizados bajo la categoría `Extreme`.⁹

9.3.1. Codificación del cuerpo + inserción de fotos

Si separamos las fotografías usando SCAN TAILOR (ver sección 6.2.6) DJVU SMALL MOD nos da la posibilidad de insertarlas en el mismo momento en que codificamos el cuerpo del documento.

Para hacerlo presionamos `(Set manual segmentation)` y configuramos las opciones como las de la figura 9.6. En particular, debemos elegir `Enable layers separation` `User-defined`, el campo `Text` debe decir `<name>.<ext>` y el campo `Background` `background\<name>.<ext>`. De esta manera reconocerá las imágenes de la carpeta `background` como fotos para insertar.

⁹ Los perfiles de cada categoría están muy bien explicados en la documentación del programa.

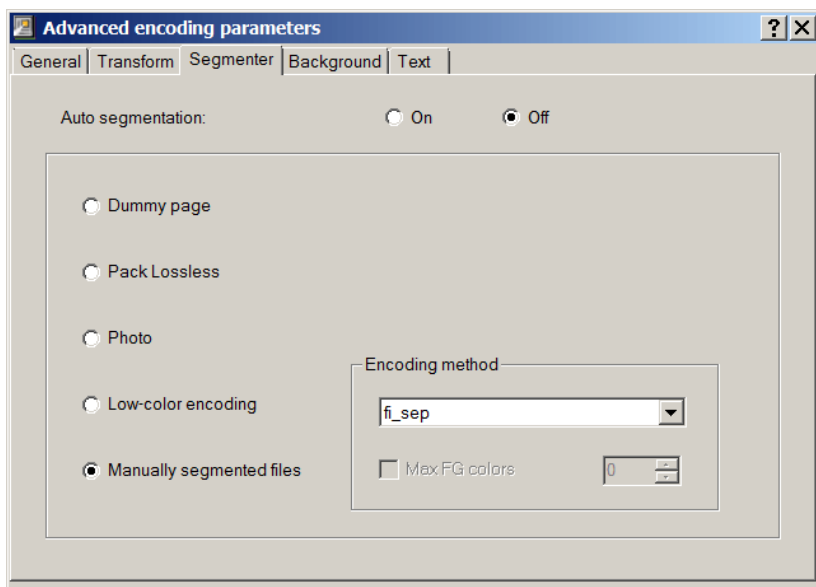


FIGURA 9.7. Pestaña **Segmenter** de las opciones de DJVU SMALL MOD.

Luego habilitamos el botón ☒ **Options** y lo presionamos. Serán de interés las pestañas **Segmenter** y **Background**.

En la primera (fig. 9.7), debemos seleccionar **Auto segmentation** ☒ **Off** ☐ **Manually segmented files**. Consulté personalmente a textsharik por las diferencias entre cada **Encoding method** y me explicó que para el caso que nos ocupa esencialmente no las hay.¹⁰ De acuerdo a la documentación, elegimos **Encoding method** **fi_sep**, que se describe como el más confiable.

En la segunda pestaña (fig. 9.8) tenemos que experimentar un poco.¹¹ En la documentación están todos los detalles, pero recomiendo

¹⁰ Textualmente dijo: «Yes, in simple cases (when foreground is B&W, and layers don't overlap) there is no difference between the manual segmentation methods. (Although encoding speed may differ.) The method selection was intended only for complex cases. I'll try to document these methods better in the next version (when I have time for it). And I'll also hide the method selection by default, as rarely-used option».

¹¹ Sugiero comenzar con los valores de la figura 9.8 y a partir de allí ir comparando.

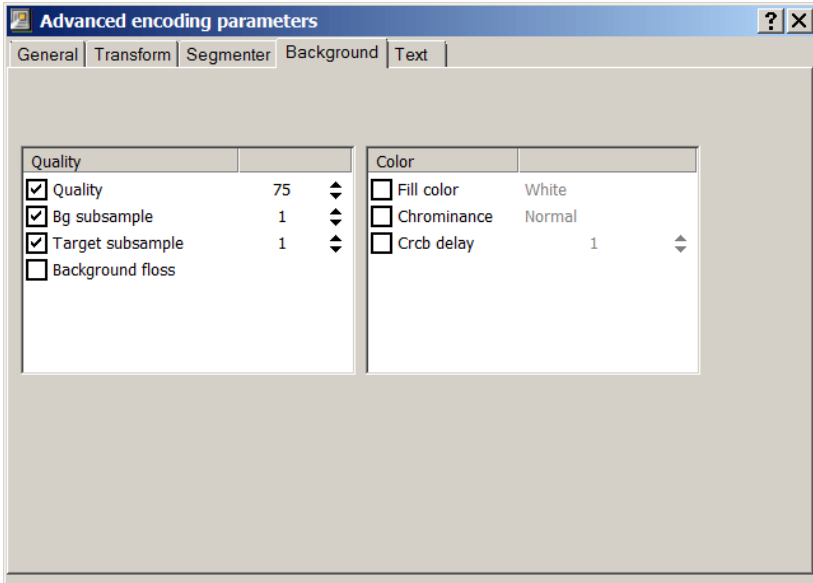


FIGURA 9.8. Pestaña **Background** de las opciones de DJVU SMALL MOD.

modificar solo las opciones del cuadro izquierdo (**Quality**).

La opción ☒Bg subsample tiene el mismo efecto que la **BSF** de la sección 9.2.4, así que recomiendo dejarla en 1 por las mismas razones que allí se explicaron. Según la documentación, ☒Target subsample sería algo así como un divisor de la calidad, por lo que también lo dejo en 1. ☐Background floss no tiene efecto en nuestro caso. Nos queda ☒Quality para probar. 75 es un valor razonable para empezar.

Una vez configuradas las opciones, presionamos **Open file(s)** y agregamos todos los archivos del cuerpo del documento. Para agregar las fotos presionamos **+** (**Add new files**) y seleccionamos todos los archivos de la carpeta **background**. Al recorrer la lista, las páginas donde se van a insertar las fotografías deberán verse como las de la figura 9.9.

Allí vemos que la página **[0014]** estará conformada por dos archivos llamados **SCAN_015.tif**: uno bitonal (tomado de **out**) y otro con las fotos (tomado de **background**). Llamaremos al archivo **BODY_INS.djvu**.

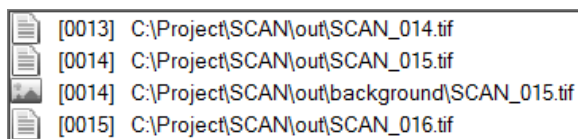


FIGURA 9.9. **Input file(s)** de DJVU SMALL MOD con imágenes bitonales y sus correspondientes con fotografías.

Para codificar las imágenes bitonales DJVU SMALL MOD usará el perfil **MyBitonal** y para las fotos las opciones explicadas más arriba.

9.3.2. Codificación de las tapas

Es igual que con DJVU SMALL (sec. 9.2.2). En este caso, obtuve un muy buen resultado usando el perfil **Lowcolor encoding**, de la categoría **Extreme**: segmentación de capas perfecta y solo 80 kB en total.

9.3.3. Unión de cuerpo y tapas

Si no queremos adaptar y usar el script 9.1 podemos unir el cuerpo y las tapas con DJVU SMALL MOD.

Para ello, elegimos **Choose operation** > **Decode DjVu** y **Output format** > **djvu**. Luego presionamos **Open pages**, seleccionamos **COVERS.djvu** y **BODY_INS.djvu** y presionamos **Abrir**. Se deberían cargar todas las páginas en la lista de archivos y la ventana debería verse como en la figura 9.10. Presionamos **Decode** y deberíamos obtener el DJVU completo, que llamaremos **SCAN_INS.djvu**.

Si por alguna razón el orden en la lista de archivos no fuera el correcto, podemos presionar **X** (**Edit files list**) y ordenarlos manualmente. Aunque si fuimos preservando la correcta numeración de los archivos esto no debería ocurrir.

9.4. Insertar la capa ocr

Para insertar la capa de texto que reconocimos tenemos el script 9.3, que mediante la ejecución del programa **djvused.exe** de DJVULIBRE

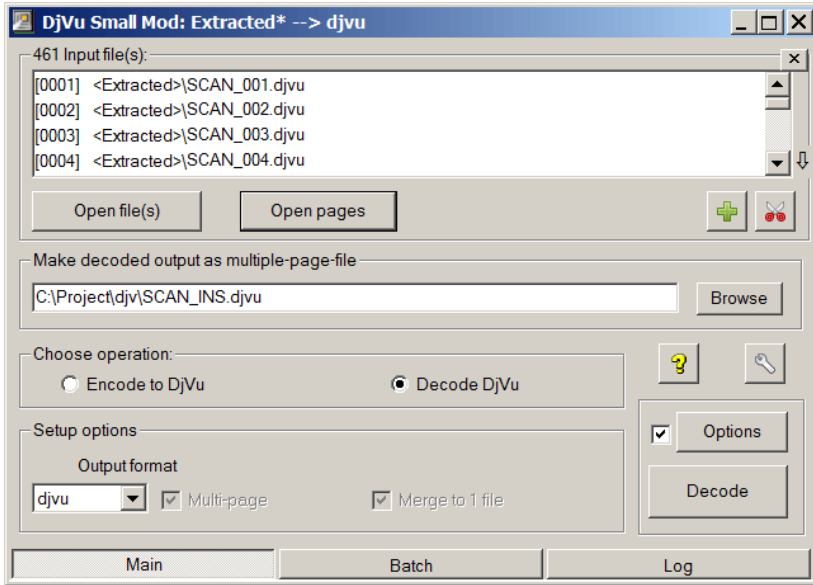


FIGURA 9.10. Uniendo el cuerpo con las tapas en DJVU SMALL MOD.

extrae la capa de texto del archivo `OCR.djvu` que creamos con FINE-READER. Luego de la búsqueda y sustitución de algunas expresiones regulares, usa el mismo programa para insertarla en el documento `SCAN_INS_OCR.djvu`.

insertOCR.py

```
# File insertOCR.dsed is the core of the whole
  ↳ operation. It does not only contain the OCR
  ↳ information, but it's also a script for djvused
  ↳ to run. This Python script creates that script,
  ↳ makes some modifications to it and then calls
  ↳ djvused to apply it, generating an OCRed djvu.
```

```
import os, re
```

insertOCR.py (cont.)

```
OCR = '../djv/OCR.djvu'
SCAN = '../djv/SCAN_INS.djvu'
SCAN_OCR = '../djv/SCAN_INS_OCR.djvu'
insertOCR = 'insertOCR.dsed'

# Delete old files (if they exist).
if os.path.isfile(SCAN_OCR):
    os.remove(SCAN_OCR)

if os.path.isfile(insertOCR):
    os.remove(insertOCR)

# Call djvused to generate a text file (the script)
↳ with the OCR information from OCR.djvu. We
↳ accomplish this by using the ">" symbol in the
↳ command line, which outputs everything to a file.
↳ There are certainly more pythonic ways to get it
↳ done.
os.system('djvused ' + OCR + ' -e output-txt > ' +
↳ insertOCR)

with open(insertOCR, 'r') as outputFile:
    content = outputFile.read()

# First, replace lines:
# select "page0001.djvu" # page X
# with:
# select X
content = re.sub(r'"page\d+\.djvu\" # page ',
↳ '', content, flags = re.M)
```

insertOCR.py (cont.)

```

# Now, delete many lines corresponding to
↳ spurious space characters. The result of
↳ not doing it can be seen by selecting
↳ text from the ABBY-created OCR.djvu.
↳ Since DjVu viewers automatically
↳ introduce a space in between words,
↳ copying text with some viewers will
↳ result un doubly spaced words. Besides, I
↳ find it more pleasing to the eye this
↳ way.
content = re.sub(r' \(\char \d+ \d+ \d+ \d+ "
↳ "\\)\n', '', content, flags = re.M)

with open(insertOCR, 'w') as outputFile:
    outputFile.write(content)
    # Append the line "save-bundled" to the end
    ↳ of the script. This line makes djvused
    ↳ save the file as SCAN_OCR.djvu when it
    ↳ reads the script.
    print('save-bundled ' + SCAN_OCR,
    ↳ file=outputFile)

# Finally, call djvused on the corrected script.
os.system('djvused ' + SCAN + ' -f ' + insertOCR)

```

SCRIPT 9.3. Inserción de la capa de OCR en documento DJVU.

9.5. Insertar marcadores

Para insertar los marcadores podemos usar la misma técnica de la sección 8.2 o correr el script 9.4. Para que funcione debe existir el archivo `insertBookmarks.txt` con el formato correcto. Al terminar debemos tener un archivo `SCAN_INS_OCR_BOOK.djvu`.

insertBookmarks.py

```
# This script uses djvused to insert the bookmarks in  
↳ the djvu file.  
  
import os, re, shutil  
  
SCAN = '../djv/SCAN_INS_OCR.djvu'  
SCAN_BOOK = '../djv/SCAN_INS_OCR_BOOK.djvu'  
bookmarks = 'insertBookmarks.txt'  
insertBookmarks = 'insertBookmarks.dsed'  
temp = 'temp.tmp'  
  
# Delete old files (if they exist).  
if os.path.isfile(SCAN_BOOK):  
    os.remove(SCAN_BOOK)  
if os.path.isfile(insertBookmarks):  
    os.remove(insertBookmarks)  
  
# Replace last TAB un each line by a space (required  
↳ by bookm).  
with open(bookmarks, 'r') as outputFile:  
    content = outputFile.read()  
    content = re.sub(r'(.+)\t(\d+)', r'\1 \2',  
        ↳ content, flags = re.M)  
with open(insertBookmarks, 'w') as outputFile:  
    outputFile.write(content)  
  
# Call bookm to generate djvused bookmark script.  
os.system('bookm ' + insertBookmarks + ' ' + temp)  
shutil.copy(temp, insertBookmarks)  
os.remove(temp)  
  
# Add set-outline to the beggining. Add . and  
↳ save-bundled to the end.  
with open(insertBookmarks, 'r') as outputFile:
```

insertBookmarks.py (cont.)

```

        content = outputFile.read()
    with open(insertBookmarks, 'w') as outputFile:
        print('set-outline', file=outputFile)
        outputFile.write(content)
        print('.', file=outputFile)
        print('save-bundled ' + SCAN_BOOK,
              ↪ file=outputFile)

# Call djvused on the bookmarking script.
os.system('djvused ' + SCAN + ' -f ' +
          ↪ insertBookmarks)

```

SCRIPT 9.4. Inserción de marcadores en documento DJVU.

9.6. Modificar etiquetas de páginas

Como explicamos en la sección 8.3 podemos modificar las etiquetas de las páginas para que coincidan con los números de página impresos en el material.

Para ello, podemos modificar el script 9.5, de forma que se adapte a las características de nuestro documento y ejecutarlo. Si todo sale bien tendremos un archivo `SCAN_INS_OCR_BOOK_PAG.djvu` con las etiquetas de página correctas.

paginate.py

```

# This script uses djvused to generate the page
↪ labels for the djvu file.

import os

SCAN = '../djv/SCAN_INS_OCR_BOOK.djvu'
SCAN_PAG = '..\djv\SCAN_INS_OCR_BOOK_PAG.djvu'
paginate = 'paginate.dsed'

```

paginate.py (cont.)

```
# Delete old files (if they exist).
if os.path.isfile(SCAN_PAG):
    os.remove(SCAN_PAG)
if os.path.isfile(paginate):
    os.remove(paginate)

# Define the page labels for the different parts.
frontMatter = ['Cover', 'FFEP']#, 'FFEP2', 'FFEP3']
romans = ['i', 'ii', 'iii', 'iv', 'v', 'vi', 'vii',
    ↪ 'viii', 'ix', 'x']#, 'xi', 'xii', 'xiii', 'xiv',
    ↪ 'xv', 'xvi', 'xvii', 'xviii', 'xix', 'xx', 'xxi',
    ↪ 'xxii', 'xxiii', 'xxiv', 'xxv', 'xxvi', 'xxvii',
    ↪ 'xxviii', 'xxix', 'xxx']
backMatter = ['RFEP', '"Back cover"', 'Spine']

# Total number of pages in the book.
total = 461;

# Calculate some indices.
fmLen = len(frontMatter) + len(romans);
pPages = total - fmLen - len(backMatter);
lastPage = total - len(backMatter);

# Generate the script and save it.
with open(paginate, 'w') as outputFile:
    for p, title in enumerate(frontMatter+romans,
        ↪ start=1):
        print('select', p, file=outputFile)
        print('set-page-title', title,
            ↪ file=outputFile)

    for p in range(1, pPages+1):
        print('select', p+fmLen,
            ↪ file=outputFile)
```

paginate.py (cont.)

```

        print('set-page-title', p,
              ↪ file=outputFile)

    for p, title in enumerate(backMatter,
                              ↪ start=lastPage+1):
        print('select', p, file=outputFile)
        print('set-page-title', title,
              ↪ file=outputFile)

    print('save-bundled ' + SCAN_PAG,
          ↪ file=outputFile)

# Call djvused on the created script.
os.system('djvused ' + SCAN + ' -f ' + paginate)

```

SCRIPT 9.5. Modificación de etiquetas de páginas en documento DJVU.

9.7. Insertar metadatos

Esta es un área del formato DJVU que aún no se encuentra demasiado desarrollada, ya que los visualizadores más populares aún no la soportan. Sin embargo, no es mala idea incluir los metadatos del material en cuestión en el documento final.

Para hacerlo, editamos los valores de interés en el script 9.6 y lo ejecutamos para obtener `SCAN_INS_OCR_BOOK_PAG_MET.djvu`. Podemos usar la herramienta EXIFTOOL para verificar que la información quedó guardada correctamente. Los campos disponibles están detallados en la propia web de EXIFTOOL [35].

insertMetadata.py

```

# This script uses djvused to insert the metadata in
↪ the djvu file.

```

insertMetadata.py (cont.)

```
import os

SCAN = '../djv/SCAN_INS_OCR_BOOK_PAG.djvu'
SCAN_MET = '../djv/SCAN_INS_OCR_BOOK_PAG_MET.djvu'
insertMetadata = 'insertMetadata.dsed'

# Delete old files (if they exist).
if os.path.isfile(SCAN_MET):
    os.remove(SCAN_MET)
if os.path.isfile(insertMetadata):
    os.remove(insertMetadata)

# Define metadata fields
fields=['author B.A. Gregory',
        'title An Introduction to Electrical
        ↳ Instrumentation and Measurement Systems',
        'subtitle A guide to the use, selection, and
        ↳ limitations of electrical instruments and
        ↳ measurement systems',
        'publisher Halsted Press; John Wiley & Sons',
        'year 1981',
        'edition 2',
        'series ',
        'isbn 0470270926',
        'isbn13 9780470270929',
        'note Ex libris Noitaenola',
        'url http://gen.lib.rus.ec/']

# Create the script.
with open(insertMetadata, 'w') as outputFile:
    print('select', file=outputFile)
    print('set-meta', file=outputFile)
    for x in fields:
        print(x, file=outputFile)
    print('.', file=outputFile)
```

insertMetadata.py (cont.)

```

print('save-bundled ' + SCAN_MET,
      ↪ file=outputFile)

# Call djvused on the generated script to save the
↪ metadata.
os.system('djvused ' + SCAN + ' -f ' +
      ↪ insertMetadata)

```

SCRIPT 9.6. Inserción de metadatos en documento DJVU.

9.8. Creación de hipervínculos

Se puede usar el DJVU EDITOR para agregar hipervínculos al documento. Estos pueden ser internos o externos.

Los internos apuntan a páginas del propio documento. Por ejemplo, se pueden crear vínculos desde las entradas de la tabla de contenidos hacia las correspondientes páginas del libro. Los externos apuntan a otros archivos o sitios web.

Su creación es muy sencilla. Se selecciona la herramienta deseada de la barra, por ejemplo **Oval Hyperlink** y se dibuja en la zona de la página donde se desea colocar. Así se abre el cuadro de diálogo de la figura 9.11. Allí rellenamos las opciones y al terminar presionamos **OK**. Una vez hayamos culminado la creación de nuestros hipervínculos guardamos el documento con **File >> Save As...** y poniendo el nombre **SCAN_INS_OCR_BOOK_PAG_MET_LINK.djvu**.

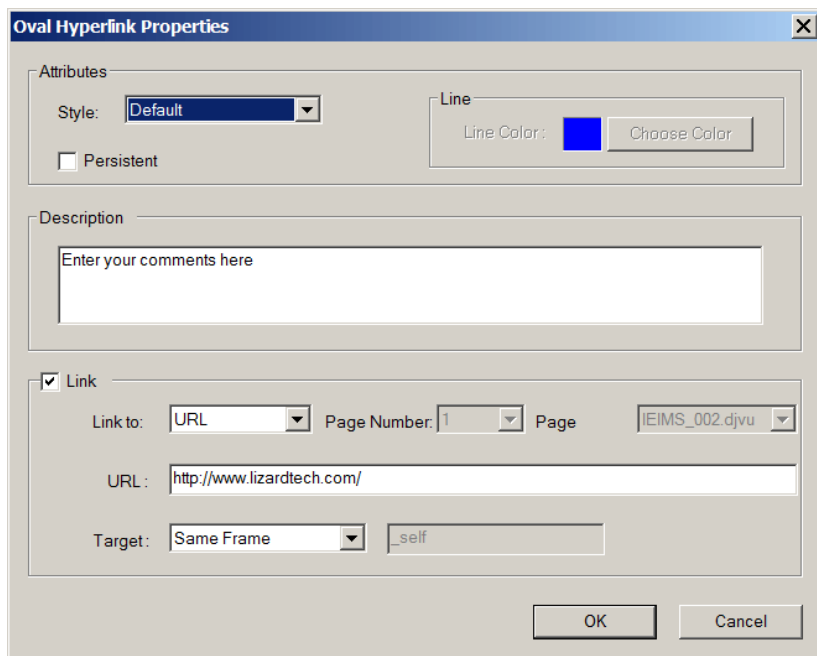


FIGURA 9.11. Menú para creación de hipervínculos en DjVu EDITOR.

Parte III

Apéndices

Apéndice A

Otras herramientas útiles

En este apéndice se mencionan otras herramientas que conviene tener a mano para este tipo de proyectos. Algunas están discontinuadas, han quedado obsoletas o han sido superadas.

DjVuToy Completa herramienta para la manipulación y conversión de archivos DJVU. Freeware para uso no comercial. Permite unir y dividir archivos; borrar, insertar o rotar páginas; convertir desde varios formatos a DJVU y viceversa; insertar y extraer la capa OCR; y analizar la estructura interna de archivos DJVU.

FreePic2Pdf Software para generar archivos PDF a partir de imágenes. Freeware para uso no comercial. Soporta variedad de formatos de imagen y diferentes opciones de compresión.

ExifTool Librería Perl y aplicación de línea de comandos para la lectura, escritura y edición de meta información en una gran variedad de tipos de archivos. Licencia GPLv1 o Artistic License.

DjVuOutline Programa para la creación y edición de marcadores en archivos DJVU. Basado en  `djvused.exe` de DJVULIBRE.

HandyOutliner Ídem DJVUOUTLINE.

pdf2djvu Herramienta de línea de comandos para crear archivos DJVU a partir de PDF. Licencia GPLv2. Existen muchas GUI para trabajar con ella.

Any2DjVu Herramienta en línea gratuita que ofrece la creación de archivos DJVU a partir de una gran variedad de formatos de origen. De los autores DJVULIBRE.

MiniDjVu Utilidad de línea de comandos para codificar y decodificar archivos DJVU en blanco y negro. La versión para Windows incluye una GUI mínima basada en *Javascript* que funciona desde Internet Explorer. Licencia GPLv2. No se actualiza desde el 2009. No la he probado, pero su autor declara que la intención es que se use para experimentar. Los comentarios en la Runet [36] son similares y hablan de que la calidad de compresión no es la mejor.

MiniDjVu Plus Interfaz gráfica para el MINIDJVU. Además agrega decodificación y codificación de imágenes a color mediante aplicaciones de DJVULIBRE. Licencia GPLv2. Herramienta experimental. No se actualiza desde el 2010.

LizardTech DjVu Solo Una de las primeras aplicaciones (en su momento prácticamente la única) para crear documentos DJVU. Gratuita para uso no comercial. Discontinuada en el año 2000 y ya superada por las otras alternativas.

Apéndice B

Conversión pix-pulgadas

Para convertir entre pixeles y pulgadas las fórmulas son un poco más sencillas, pues no incluyen el factor 2,54:

$$\text{pix} = \text{in} \times \text{dpi} \qquad \text{in} = \frac{\text{pix}}{\text{dpi}}$$

El cuadro 6.1 se convierte en el cuadro B.1.

(600 dpi)	ancho	alto	espesor
in	6.1	9.3	1.0
pix	3660	5580	600

CUADRO B.1. Dimensiones del libro a escanear (en in-pix).

Sobre el material de ejemplo

Como se explica al comienzo del capítulo 4, esta guía fue elaborada tomando como ejemplo la digitalización del libro *An Introduction to Electrical Instrumentation and Measurement Systems*, de B. A. Gregory [27]. Se trata de un ejemplar tapa blanda, compuesto de texto, ecuaciones y diagramas monocromáticos, así como de varias fotografías en escala de grises (semitonos [37]).

Lo elegí por varias razones. La primera es que tengo un ejemplar físico en mi biblioteca y me interesaba obtener un respaldo del mismo en formato digital. Por otra parte, es un caso realista de lo que se podría querer digitalizar y por ello me pareció apropiado usarlo como ejemplo en esta guía. Además, la editorial *Springer* tiene a la venta la versión digital de este libro,¹ producida por métodos similares, por lo que existe un punto de referencia con el cual comparar nuestros resultados.

Los documentos en PDF² y DJVU³ creados junto con esta guía, así como el PDF⁴ de *Springer* se pueden descargar en *Library Genesis* [38].

¹ En realidad se trata de una reimpresión posterior bajo otra editorial y con algunas correcciones, aunque muy parecida.

² MD5 (mi PDF): 646923F422762352F3E47360952BE03C.

³ MD5 (mi DJVU): C34ABFC96152F9B881CCD06593A7CD36.

⁴ MD5 (PDF Springer): 7A43F6FAE8ABB7450FD3A2978D5E295C.

Alternativas para procesamiento de imágenes

Si no queremos procesar independientemente el texto y las fotografías de nuestro material tenemos algunas opciones.

Advertencia No recomiendo ir por esta ruta puesto la calidad resultante es realmente baja. Los explico aquí porque son métodos que por desconocimiento usé hace muchos años, pero sobre todo como ejemplo de lo que *no se debe hacer*.

Primero que nada, no debemos activar ☒ Split pages de SCAN TAILOR o, en su defecto, no hacer la sustitución que se explica al principio de la sección 7. El procedimiento de limpieza es el que se describe en la sección 6.

Antes de comenzar con el OCR tenemos dos alternativas: convertimos las imágenes en bitonales o las procesamos más adelante.

Usando GIMP podemos transformar las páginas que tienen fotografías (especialmente en escala de grises) en bitonales. Para ello abrimos la imagen y vamos a **Image > Mode > Indexed...**. Se abrirá el menú de la figura D.1. Allí elegimos ☒ Use black and white (1-bit) palette y en

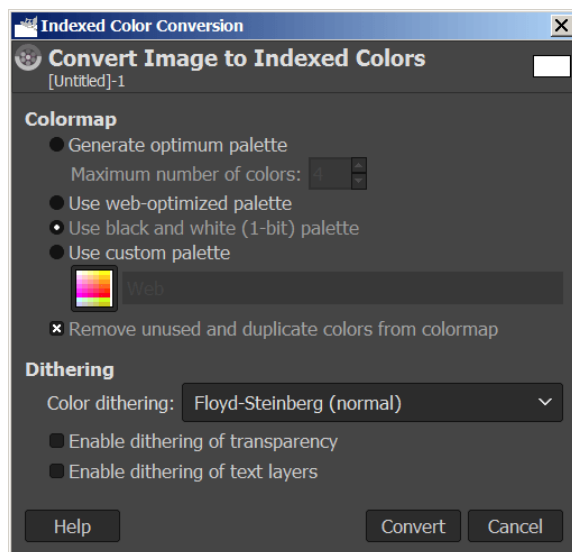


FIGURA D.1. Menú **Convert Image to Indexed Colors** de GIMP.

Color dithering elegimos alguna de las opciones: None, Floyd-Steinberg (Normal) o Reduced color bleeding) o Positioned. Normalmente None da los peores resultados.

De esta manera, se obtiene una reducción importante del tamaño a costa de la calidad. En el ejemplo de la página 128, se pasó de un archivo de 10,5 MB a uno de 525 kB. En la figura D.2 se ilustra un poco el efecto que produce en una fotografía la conversión a bitonal. Por supuesto que estará acentuado en fotos a color.

De proceder así, todas nuestras imágenes serán bitonales (a excepción tal vez de las tapas). Así las cargaremos en FINEREADER y luego podremos codificarlas en DJVU usando el perfil MyBitonal sin discriminación.

D.1. Procesamiento posterior

La otra alternativa es no tocar nada hasta el momento de codificar. En el caso del PDF, deberemos optimizar el archivo para bajar un poco

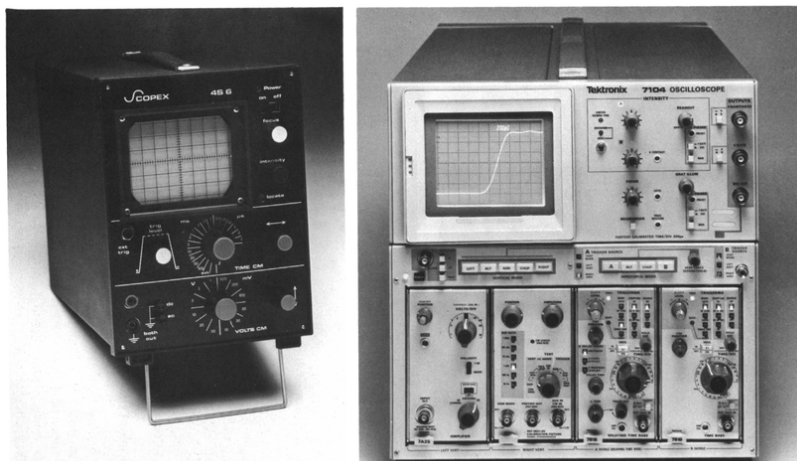


Figure 2.52. Cathode ray oscilloscopes: (a) simple, (b) complex (courtesy of Scopex Ltd and Tektronix (U.K.) Ltd)

(a) Original.

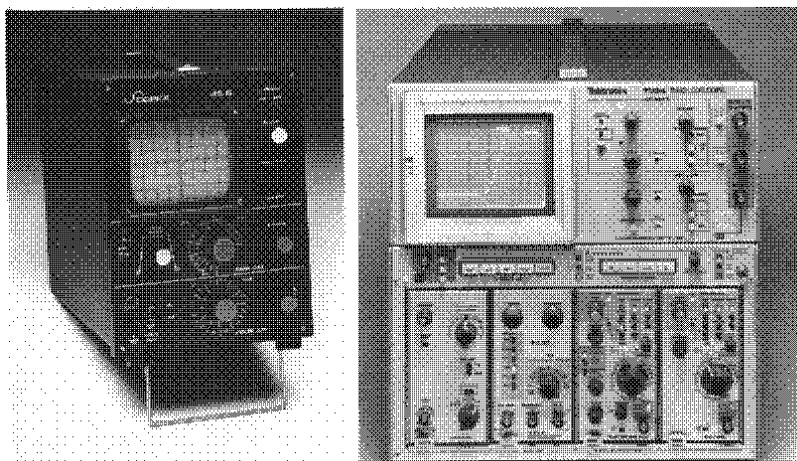


Figure 2.52. Cathode ray oscilloscopes: (a) simple, (b) complex (courtesy of Scopex Ltd and Tektronix (U.K.) Ltd)

(b) Bitonal (Positioned).

FIGURA D.2. Comparación entre original y binarizado. El efecto ha sido acentuado por claridad. En la práctica no será tan notorio.

el tamaño como se explica en 8.5. La desventaja es que las páginas bitonales se codificarán usando el algoritmo JBIG, mientras que las que contengan fotos usarán JPEG. Este método de compresión no es el más adecuado para el texto, por lo que aparecerán los característicos artefactos de la figura D.3 alrededor del mismo.

Para el DJVU confiaremos en que el algoritmo de segmentación automática del perfil `MyScan` o similar haga un buen trabajo de separar el texto de las imágenes. Si esto ocurre el resultado será de buena calidad, salvo quizás por el tamaño del archivo resultante.

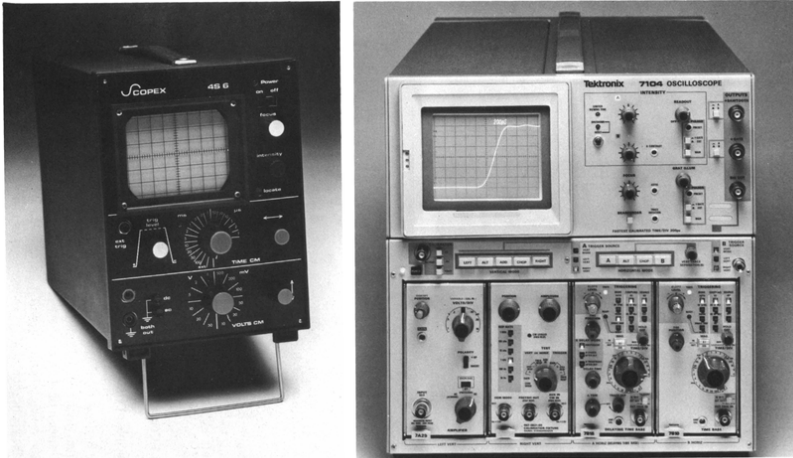


Figure 2.52. Cathode ray oscilloscopes: (a) simple, (b) complex (courtesy of Scopex Ltd and Tektronix (U.K.) Ltd)

(a) Original.

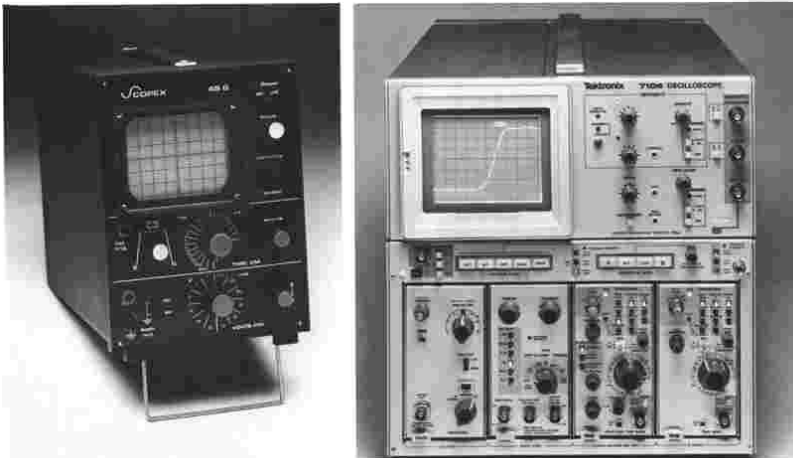


Figure 2.52. Cathode ray oscilloscopes: (a) simple, (b) complex (courtesy of Scopex Ltd and Tektronix (U.K.) Ltd)

(b) JPEG.

FIGURA D.3. Comparación entre original y JPEG. El efecto ha sido acentuado por claridad. En la práctica no será tan notorio.

Apéndice E

Materiales complejos

Para los materiales impresos que no sean como los descritos al principio de la sección 1.2 podemos usar un procedimiento similar al de esta guía.

Para el proceso de escaneo se pueden seguir las indicaciones de la sección 5.

La limpieza y ajuste se simplificarán a costa de no corregir prácticamente nada. En el paso 6 de SCAN TAILOR debemos elegir **Mode** **> Color / Grayscale**, que aplicará todos los filtros pero no binarizará nada. De esta forma obtendremos las imágenes recortadas y desinclinadas, con márgenes que deberemos seleccionar en cada caso.

Esas imágenes son las que procesaremos con FINEREADER, de la misma forma que se explica en 7 (o en el apéndice F).

Para la codificación tanto del PDF como del DJVU ignoraremos los pasos de inserción de imágenes.

En el caso del PDF, tendrá mucho peso el paso 8.5. Recomendando experimentar con las diferentes opciones del menú **Images**.

Para el DJVU no podremos utilizar el perfil **MyBitonal**. Deberemos hacer ensayos con **MyScan** o **MyPhoto**, y preferentemente con los perfiles de la categoría **Extreme** de DJVU SMALL MOD. Este programa también ofrece métodos de segmentación automática y semiautomática que no he probado pero que se ven muy prometedores. También existen otras

aplicaciones para realizar este tipo de tareas [39].

Una vez codificado el archivo, los pasos de insertar la capa OCR, marcadores, metadatos, etc. son idénticos.

Ocr simplificado

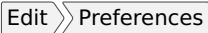
Una al FINEREADER para incluir el OCR en nuestros documentos usar ACROBAT DC y DJVU EDITOR.

Este método tiene por ventaja la rapidez, porque es totalmente automatizado. En cambio presenta varias desventajas. La primera es que vamos a usar un software distinto para cada formato, por lo que la capa de texto seguramente difiera de uno a otro.¹ Tampoco podremos hacer ningún tipo de corrección² ni reconocer texto rotado.

F.1. Con Acrobat DC

Con ACROBAT DC tenemos el así llamado *Image Conversion Plug-in*.³ Hay un par de formas diferentes de proceder y que utilizan cuadros de diálogo similares. Voy a describir la que considero mejor.

F.1.1. Configuración previa

Lo primero que haremos será configurar las opciones de codificación de ACROBAT DC (fig F.1). Para ello vamos a 

¹ No sé como exportar la capa OCR de un formato e insertarla en otro.

² ACROBAT DC sí permite corregir, pero considero a la herramienta poco útil.

³ Anteriormente llamado *Paper capture plugin*.

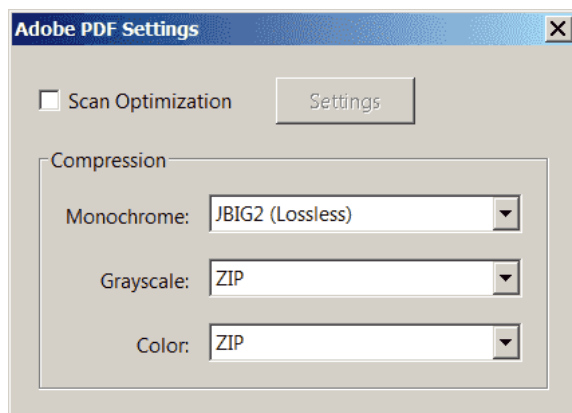


FIGURA F.1. Menú **Edit Settings** del formato TIFF en ACROBAT DC.

(**Ctrl** + **K**). Allí elegimos **Categories** > **Convert To PDF** y presionamos **Converting To PDF** > **TIFF** > **Edit Settings**. Deberá abrirse el cuadro de la figura F.1.

Sin bien aquí se pueden elegir varias opciones de compresión, no vamos a usar ninguna en esta etapa. Ponemos todas las opciones de **Color Management** en **Preserve embedded profiles**. Además elegimos **JBIG2 (Lossless)**, **ZIP** y presionamos **OK**. De momento solo queremos crear el documento base. La reducción de tamaño la realizaremos más adelante.

Ahora vamos a **File** > **Create** > **Combine Files...** y presionamos **Options**. En el cuadro que aparece (fig. F.2) desactivamos todas las casillas, presionamos **File Size** > **Default File Size**⁴ y luego **OK**.

F.1.2. Creación del documento base

En este momento agregamos todos los archivos de la carpeta **out**, ya sea seleccionando y arrastrando o usando **Add Files**. Luego presionamos **Combine**. ACROBAT DC se tomará algo de tiempo y creará un

⁴ Como se explica en el tooltip que se despliega al pasar con el ratón por encima, las opciones estarán controladas por el panel de preferencias, que acabamos de configurar.

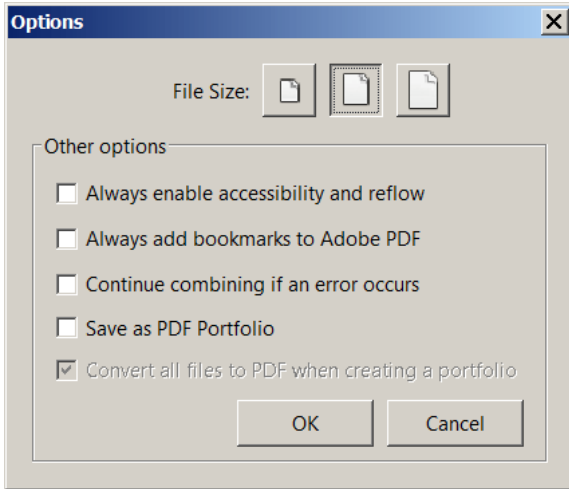


FIGURA F.2. Menú **Combine Files into a Single PDF...** » **Options** en ACROBAT DC.

PDF con todas las imágenes. Lo guardaremos en la carpeta djv con el nombre SCAN.pdf.

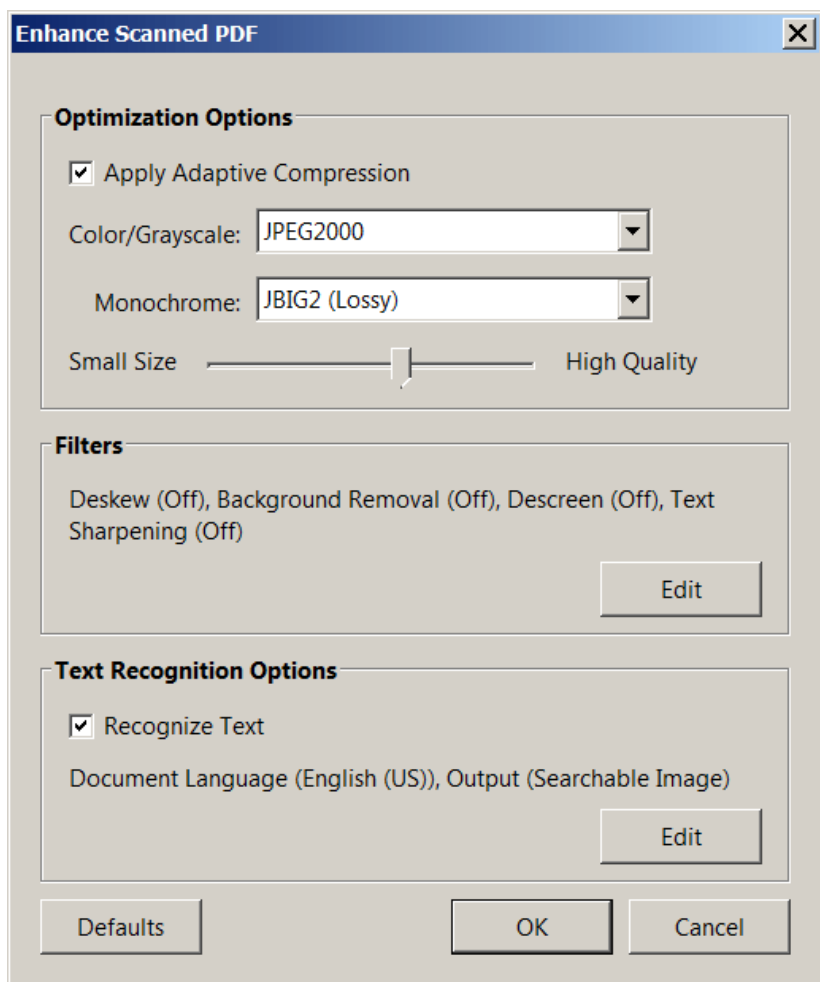
F.1.3. Optimización y ocr

Llegó el momento de optimizarlo y agregar la capa OCR. Vamos a **View** » **Tools** » **Enhance Scans** » **Open** para abrir la barra de herramientas. Luego presionamos **Enhance** » **Scanned Document** » **Settings** para abrir el cuadro de la figura F.3

En **Optimization Options** sugiero activar ☒ **Apply Adaptive Compression**. Esta opción hace una segmentación automática similar a del formato DJVU ya explicada. Luego tenemos opciones de compresión para imágenes que se explican a continuación.

Las bitonales se pueden comprimir en **JBIG2** (**Lossy** o **Lossless**) o **CCITT Group 4** (no recomendado). Las que contengan algún elemento a color o en escala de grises en **JPEG**, **JPEG2000** o **ZIP**. Con la barra deslizante ajustamos la calidad global del documento.

En **Filters** recomiendo desactivar todas las correcciones, poniéndolas

FIGURA F.3. Menú **Enhance Scanned PDF** > **Settings** en ACROBAT DC.

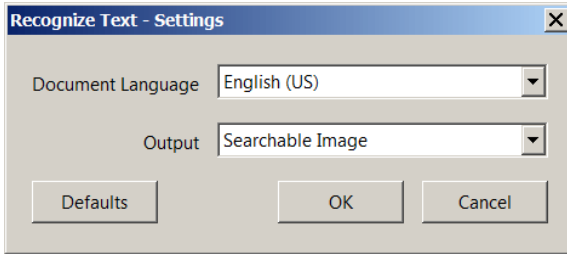


FIGURA F.4. Menú **Text Recognition Options** » **Edit** en ACROBAT DC.

en **Off**. Análogamente al trabajo con FINEREADER, ya nos encargamos de preparar el documento con SCAN TAILOR, por lo que no queremos ahora que se corrija más nada (y menos automáticamente y sin posibilidad de revisión).

Por último, en **Text Recognition Options** activamos ☒ **Recognize Text** y presionamos **Edit**. En el cuadro de dialogo que aparece (fig. F.4) tenemos para elegir el idioma (**Document Language**) y un parámetro algo más misterioso llamado **Output** con dos opciones: **Searchable Image** y **Editable Text and Images**.

La primera es equivalente a la opción **Text under the page image** de FINEREADER. Las imágenes se comprimen con las **Optimization Options** elegidas y la capa de texto se inserta de manera invisible por debajo de estas.

A la segunda se le llamó originalmente *ClearScan* [40]. Lo que hace es tratar de generar una fuente sintética a partir del texto reconocido y utilizarla para reemplazar las imágenes de cada carácter. De esta manera pretende reducir el tamaño y obtener un resultado visualmente superior. Se ilustra un ejemplo en la figura F.5

El problema es que depende de que el OCR detecte correctamente los caracteres para funcionar bien, porque de lo contrario no sabe con cual reemplazarlos. Cuando esto ocurre usa una versión de baja resolución del símbolo que no reconoció, lo que da un resultado desparejo. En otras ocasiones corrompe algunas figuras. Otro detalle es que al activarla y dar **OK** ACROBAT DC nos informa: *Compression settings will not be applied with selected Text Recognition output*. Es decir, que ignorará todas nuestras opciones anteriores y ajustará, corregirá y comprimirá lo que



(a) Imagen original.



(b) Imagen después de `ClearScan`.

FIGURA F.5. Resultado la opción `Editable Text and Images`/`ClearScan` sobre texto reconocido.

quiera y cuanto quiera.

La opción `Editable Text and Images`/`ClearScan` tiene encantadas a muchas personas. Yo no soy fan. La menciono por su amplio uso. Dependerá de cada uno si usarla o ir por la vieja y querida `Searchable Image`.

F.2. Con DjVu Editor

Agregar la capa al DJVU es bastante más sencillo. En lugar de aplicar el script de la sección 9.4 (si estamos aquí es que no existe), vamos a usar DJVU EDITOR para generar el OCR.

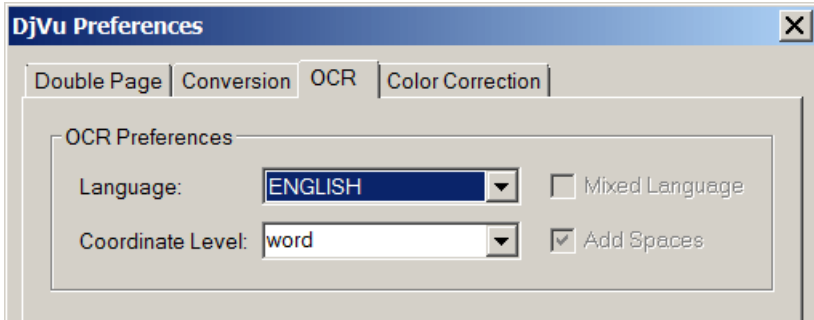


FIGURA F.6. Preferencias de OCR en DJVU EDITOR.

Abrimos el archivo SCAN_INS.djvu con el programa y vamos a Edit >> Preferences >> OCR. En Language >> (Idioma) elegimos el idioma correspondientes y luego Coordinate Level >> word. Presionamos OK y vamos a Tools >> OCR >> OCR Entire Document para que haga el reconocimiento. Se tomará un rato para hacerlo. Al terminar lo guardamos con el nombre SCAN_INS_OCR.djvu haciendo File >> Save As... y seguimos con la sección 9.5.

Estructura del proyecto de FineReader 14

En la figura G.1 se presenta un diagrama del árbol de directorios que se genera cuando guardamos un proyecto de FINEREADER v14.

El software crea una carpeta para cada página que vamos agregando. Esas carpetas son numeradas sucesivamente utilizando cuatro dígitos: 0001, 0002, ... Dentro de la carpeta se crean archivos que guardan información sobre esa página, las transformaciones que se le hicieron, su miniatura, etc, pero lo más importante es que para las páginas en blanco y negro se crea un archivo `bwPage.frd`, que no es más que el TIFF original copiado y pegado allí.

Si FINEREADER no tiene esa página abierta, podemos reemplazar ese archivo por un TIFF con el mismo nombre y el software creará que se trata de la imagen original. Esto funciona siempre que el nuevo TIFF también sea en blanco y negro y que además tenga las mismas dimensiones que el anterior.

Algo parecido ocurre con las páginas a color, solo que estas se guardan divididas en sus componentes YCbCr [41]: `grayComponent.frd` guarda la componente de luma, `cbComponent.frd` la de crominancia diferencia de azul y `crComponent.frd` de crominancia diferencia de rojo. No he hecho pruebas, pero estimo que si queremos reemplazar la imagen podemos hacerlo al igual que con la páginas en blanco y

negro, solo que habrá que generar primero estas tres componentes para reemplazar, tal vez usando *ImageMagick*.

Este reemplazo permite corregir con SCAN TAILOR errores encontrados en la etapa de OCR sin tener que usar el cuadro de diálogo para reabrir la página. La ventaja de esto es que se mantiene la coherencia de la numeración: si usáramos el diálogo para reabrir, FINEREADER en realidad eliminaría la carpeta correspondiente y crearía una nueva al final de la lista con la página reabierta. Por ejemplo, si nuestro material tuviera 461 páginas y decidiéramos reabrir la número 100, FINEREADER eliminaría la carpeta 0100 y crearía la 0462, donde colocaría la nueva página.

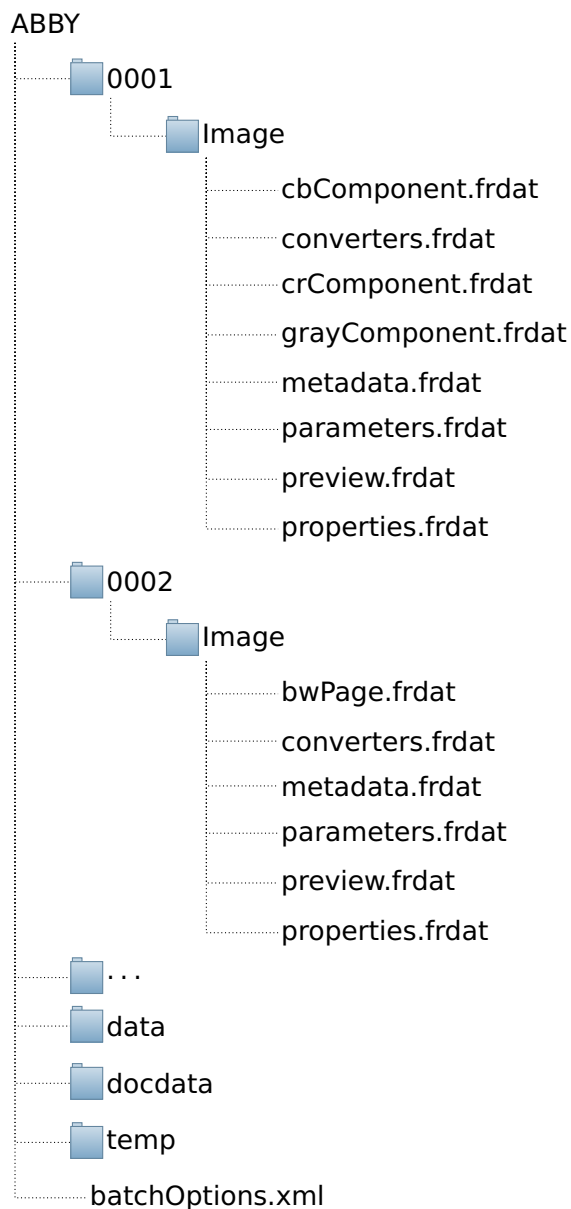


FIGURA G.1. **Árbol de directorios del proyecto de FINEREADER.**

Referencias

- [1] monday2000. *Creación de libros en formato electrónico a partir de libros en papel (en formato DjVu)*. Ruso. 2011. URL: <http://djvu-soft.narod.ru/>.
- [2] Wikimedia Commons. *Creating a DjVu file*. Inglés. 2019. URL: https://commons.wikimedia.org/wiki/Help:Creating_a_DjVu_file.
- [3] Vladimir T.T. *Tutorial para escanear y compartir 1.07*. Trad. por A. y Armando. 2008. URL: <https://es.slideshare.net/janeko/como-crear-unebookescaneadodecalidad>.
- [4] twdragon. *Creación de libros electrónicos a partir de escaneos: ¿cómo obtener DjVu o PDF de un libro de papel?* Ruso. 2009. URL: <https://rutracker.org/forum/viewtopic.php?t=2160930>.
- [5] monday2000. *Cómo hacer un libro de DjVu*. Ruso. 2008. URL: <http://www.djvu-soft.narod.ru/scan/123.htm>.
- [6] DjVu.org. *What is DjVu, and what are the secrets behind DjVu's superior performance?* Inglés. URL: <http://djvu.org/resources/whatisdjvu.php>.
- [7] ISO/IEC 16485:2000. *Information technology – Mixed Raster Content (MRC)*. Ed. por ISO/IEC JTC 1/SC 29. ISO, 2000.
- [8] Celartem/Lizardtech *DjVu Reference. DjVu v3*. Inglés. Lizardtech, 2005. URL: <http://djvu.org/docs/DjVu3Spec.djvu>.

- [9] ISO 32000-1:2008. *Document management – Portable document format – Part 1: PDF 1.7*. Ed. por ISO/TC 171/SC 2. ISO, 2008.
- [10] James Rile. *The Future of DjVu*. Inglés. 2004. URL: <https://djvufuture.blogspot.com/>.
- [11] DjVu.org. *DjVu Downloads & Resources*. Inglés. URL: <http://djvu.org/resources/>.
- [12] Wikisource. *Help:DjVu files*. Inglés. 2019. URL: https://en.wikisource.org/wiki/Help:DjVu_files.
- [13] Wikisource. *DjVu vs. PDF*. Inglés. URL: https://en.wikisource.org/wiki/Wikisource:DjVu_vs._PDF.
- [14] Jesse Tokar. *PDF vs. DjVu. Pros y contras*. URL: <https://www.print-driver.com/stories/pdf-vs-djvu>.
- [15] *DjVu Technology*. URL: https://www2.cuminas.jp/en/about_djvu.
- [16] Leon Bottou. *DjVu: Scanned Documents on the Web*. URL: https://www2.cuminas.jp/en/about_djvu.
- [17] Yann LeCun. *Yann's DjVu Page*. URL: <http://yann.lecun.com/ex/djvu/index.html>.
- [18] Wikipedia. *Variable de entorno*. URL: https://es.wikipedia.org/wiki/Variable_de_entorno.
- [19] slhck. "What are PATH and other environment variables, and how can I set or use them?" Inglés. En: *Stack Exchange* (2011). URL: <https://superuser.com/questions/284342/what-are-path-and-other-environment-variables-and-how-can-i-set-or-use-them>.
- [20] Leon Bottou y col. *DjVu Licensing*. Inglés. URL: <http://djvu.sourceforge.net/licensing.html>.
- [21] LizardTech Inc. *LizardTech's Statement on DjVu Open Source Licensing*. Inglés. 2001. URL: <http://djvu.sourceforge.net/lti-licensing.html>.
- [22] Wikipedia. *Image scanner*. Inglés. URL: https://en.wikipedia.org/wiki/Image_scanner.
- [23] Daniel Reetz. *Comunidad DIY Book Scanner*. Inglés. URL: <https://www.diybookscanner.org/>.

-
- [24] Ijon Berlin. *Libreflip*. Inglés. URL: <https://libreflip.org/>.
- [25] Dany Qumsiyeh. *Linear Book Scanner*. Inglés. URL: <https://linearbookscanner.org/>.
- [26] Kevin Vaughan. *Compare CCD vs CIS Scanner Technologies*. Inglés. 2019. URL: <https://www.tavco.net/wide-format-plotter-scanner-blog/bid/107329/compare-ccd-vs-cis-scanner-technologies>.
- [27] B. A. Gregory. *An Introduction to Electrical Instrumentation and Measurement Systems. A guide to the use, selection, and limitations of electrical instruments and measurement systems*. 2.^a ed. Macmillan, 1981. ISBN: 978-0-333-29384-3. DOI: [10.1007/978-1-349-16482-0](https://doi.org/10.1007/978-1-349-16482-0).
- [28] Wayne Fulton. *A few scanning tips*. Inglés. 2019. URL: <https://www.scantips.com/>.
- [29] Tulon. *Scan Tailor project's Wiki*. Inglés. 2014. URL: <https://github.com/scantailor/scantailor/wiki>.
- [30] Wikipedia. *Page orientation*. Inglés. URL: https://en.wikipedia.org/wiki/Page_orientation.
- [31] *JavaScript for Acrobat API Reference*. Adobe, 2015. URL: https://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/AcrobatDC_js_api_reference.pdf.
- [32] *Developing Acrobat Applications Using JavaScript*. Adobe, 2015. URL: https://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/AcrobatDC_js_developer_guide.pdf.
- [33] Wikipedia. *PDF/A*. URL: <https://es.wikipedia.org/wiki/PDF/A>.
- [34] Cptn_Cook. 17. *¿Por qué no usar un diccionario grande para todo el libro al codificar en DJVU?* Ruso. 2011. URL: <http://www.djvu-soft.narod.ru/soft/basic.htm#17>.
- [35] *DjVu Tags*. 2019. URL: <https://www.sno.phy.queensu.ca/~phil/exiftool/TagNames/DjVu.html>.
- [36] Wikipedia. *Runet*. URL: <https://es.wikipedia.org/wiki/Runet>.

- [37] Wikipedia. *Semitono*. URL: [https://es.wikipedia.org/wiki/Semitono_\(impresi%C3%B3n\)](https://es.wikipedia.org/wiki/Semitono_(impresi%C3%B3n)).
- [38] *Library Genesis*. 2019. URL: <http://gen.lib.rus.ec/>.
- [39] Jakub Wilk. *didjvu*. 2018. URL: <http://jwilk.net/software/didjvu>.
- [40] Rick Borstein. *Better PDF OCR. ClearScan is smaller, looks better*. URL: http://blogs.adobe.com/acrolaw/2009/05/better_pdf_ocr_clearscan_is_smal/.
- [41] Wikipedia. *YCbCr*. URL: <https://es.wikipedia.org/wiki/YCbCr>.

Lista de figuras

2.1. Dos ejemplos de <i>book scanners</i>	15
2.2. Archivist. Desarrollado por Daniel Reetz con ayuda de la comunidad <i>DIY Book Scanner</i>	15
2.3. OpticBook 3800, de Plustek.	16
2.4. Escáneres con ADF.	17
3.1. Árbol de directorios del proyecto.	19
5.1. Ajustes de escaneo del IRFANVIEW.	28
5.2. Ventana principal del EPSON Scan. Se previsualizó una página representativa.	29
5.3. La misma imagen que en la fig. 5.2 pero habiendo seleccionado la opción <code>Ninguna</code>	30
5.4. Distorsiones y sombras sobre el margen interior.	33
5.5. Distorsiones trapezoidal y por movimiento.	35
5.6. Transparentación sobre texto y diagramas. (Se ha exagerado un poco para que se distinga con claridad.)	36
5.7. Detalle de transparencias.	37
6.1. Cuadro de diálogo <code>New Project</code> , de SCAN TAILOR.	42
6.2. Ventana principal de SCAN TAILOR. Primer paso: Fix orientation.	43

6.3. Segundo paso: Split pages.	44
6.4. Error en el paso 2: Split pages.	45
6.5. Tercer paso: Deskew.	46
6.6. Error en el paso 4: Select content.	46
6.7. Caracteres cortados.	48
6.8. Filtro <i>despeckle</i> frente a puntos demasiado grandes.	49
6.9. Filtro <i>despeckle</i> frente a motas pegadas al texto.	49
6.10. Efecto indeseado del filtro <i>despeckle</i>	50
6.11. Sexto paso: Output. Se seleccionó la pestaña Picture zones	52
6.12. Selección manual de la imagen en pestaña Picture Zones	53
6.13. Binarización de un dibujo problemático.	55
6.14. Quinto paso: Márgenes.	56
6.15. Detalle de las guías para ajustar márgenes.	59
6.16. Página apaisada y sus contiguas en el documento final.	61
6.17. Efectos de retoque en la tapa del libro. Se pasó de un TIFF de 56 MB a uno de 686 kB con similar apariencia.	63
7.1. Opciones de FINEREADER. Pestaña Image Processing	67
7.2. Opciones de FINEREADER. Pestaña Languages	68
7.3. Opciones de FINEREADER. Creando un nuevo idioma.	69
7.4. Opciones de FINEREADER. Pestaña Format Settings » PDF	71
7.5. Opciones de FINEREADER. Pestaña Format Settings » DjVu	72
7.6. Ventana principal de FINEREADER una vez cargadas las páginas.	73
7.7. Herramientas de edición de zonas de FINEREADER.	73
7.8. Opciones de FINEREADER. Pestaña Areas and Text	76
7.9. Dos manchas no detectadas durante el paso de limpieza y guion dejado fuera del área de reconocimiento.	76
7.10. Edición del alfabeto.	78
7.11. Pestaña Area Properties de FINEREADER.	79
7.12. Algunas erratas en el libro de ejemplo.	81
7.13. Opciones de FINEREADER. Pestaña OCR.	83
7.14. Entrenamiento y creación de la ligadura «fl» en FINEREADER.	84
8.1. Consola de <i>Adobe Javascript</i> lista para insertar imágenes.	92
8.2. Formato de marcadores y resultado en el documento.	93
8.3. Ventana del PDGCNTEEDITOR. Se presionó ±	94
8.4. Ventana de numeración de páginas.	97

8.5.	PDF Optimizer de ACROBAT DC. Menú Images.	98
8.6.	Menú Preflight de ACROBAT DC, pestaña Standards.	100
9.1.	Perfiles personalizados para crear archivos DJVU.	102
9.2.	Ventana principal de DJVU SMALL.	104
9.3.	Ventana principal de DJVU IMAGER lista para convertir.	107
9.4.	Ventana de opciones de DJVU IMAGER.	108
9.5.	Ventana principal de DJVU SMALL MOD.	110
9.6.	Ventana de configuración de segmentación manual de DJVU SMALL MOD.	111
9.7.	Pestaña Segmenter de las opciones de DJVU SMALL MOD.	112
9.8.	Pestaña Background de las opciones de DJVU SMALL MOD.	113
9.9.	Input file(s) de DJVU SMALL MOD con imágenes bitonales y sus correspondientes con fotografías.	114
9.10.	Uniendo el cuerpo con las tapas en DJVU SMALL MOD.	115
9.11.	Menú para creación de hipervínculos en DJVU EDITOR.	124
D.1.	Menú Convert Image to Indexed Colors de GIMP.	134
D.2.	Comparación entre original y binarizado. El efecto ha sido acentuado por claridad.	135
D.3.	Comparación entre original y JPEG.	137
F.1.	Menú Edit Settings del formato TIFF en ACROBAT DC.	142
F.2.	Menú Combine Files into a Single PDF... Options en ACROBAT DC.	143
F.3.	Menú Enhance Scanned PDF Settings en ACROBAT DC.	144
F.4.	Menú Text Recognition Options Edit en ACROBAT DC.	145
F.5.	Resultado la opción Editable Text and Images/ClearScan sobre texto reconocido.	146
F.6.	Preferencias de OCR en DJVU EDITOR.	147
G.1.	Árbol de directorios del proyecto de FINEREADER.	151

Lista de cuadros

6.1. Dimensiones del libro a escanear (en cm-pix).	40
6.2. Márgenes ingresados para las distintas páginas (en mm). . .	57
B.1. Dimensiones del libro a escanear (en in-pix).	129

Lista de scripts

6.1. Rotación de las páginas.	41
6.2. Ajuste de páginas apaisadas.	62
7.1. Reemplazo de imagen bitonal en FINEREADER 14.	77
8.1. Inserción de fotos usando QPDF.	89
8.2. Inserción de fotos usando ACROBAT DC.	91
9.1. Unión del cuerpo con las tapas.	105
9.2. Separación de las fotos creadas con DJVU SMALL.	109
9.3. Inserción de la capa de OCR en documento DJVU.	117
9.4. Inserción de marcadores en documento DJVU.	119
9.5. Modificación de etiquetas de páginas en documento DJVU.	121
9.6. Inserción de metadatos en documento DJVU.	123

ToDo

Estos son cambios o adiciones que me falta hacer.

- conclusiones para cada capítulo
- algo sobre los warnings de FINEREADER?
- renombrar salida de ST en caso de que haya que dividir páginas escaneadas?
- alguna nota sobre la propiedad intelectual?
- Investigar este [script](#) de Python.
- homogeneizar todo el tono del documento: haremos, haré, haga?
- agregar una sección sobre como trabajar con DJVULIBRE?
- uniformizar los títulos. ej: insertar fotos vs. inserción de fotos?
- ordenar la bibliografía?
- Crear un script para agregar los metadatos con EXIFTOOL.